A journey through optimization: from global to discrete optimization and back.

Bernardetta Addis

UNIVERSITÉ DE LORRAINE

Ecole doctorale IAEM Lorraine

A journey through optimization: from global to discrete optimization and back

THÈSE

soutenue le 12 décembre 2018

pour l'obtention d'une

Habilitation de l'Université de Lorraine
(mention informatique)

par

Bernardetta Addis

Composition du jury:

Pr. Jean-Charles BILLAUT	Université de Tours	
Pr. Bernard FORTZ	Université Libre de Bruxelles	(rapporteur)
Pr. Martine LABBÉ	Université Libre de Bruxelles	
Pr. Ammar OULAMARA	Université de Lorraine	
Pr. Laura PALAGI	Sapienza Università di Roma	(rapporteur)
Pr. Alain QUILLIOT	Université Clermont Auvergne	(rapporteur)

Laboratoire Lorrain de Recherche en Informatique et ses Applications
UMR 7503

To my family and friends
that accompanied me along this not always easy journey.

Alla mia famiglia e ai miei amici
che mi hanno accompagnato lungo questo non sempre facile viaggio.

À ma famille et à mes amis
qui m'ont accompagné dans ce pas toujours facile voyage.

Bernardetta

Résumé

Ce manuscrit est le résultat de ma volonté, d'une part, de résumer mon parcours de recherche et, de l'autre, de me projeter dans l'avenir. L'exercice en question s'avère une tâche difficile principalement pour deux raisons : premièrement, je voulais écrire un document accessible à l'ensemble de chercheurs en informatique (en supposant quand même quelques connaissances de base en optimisation), et deuxièmement, bien que j'ai travaillé toujours dans le domaine de l'optimisation, mon parcours peut être considéré plutôt atypique. En effet, en relativement peu de temps, je suis passé du domaine de l'optimisation globale (non linéaire, non convexe) à celui de l'optimisation discrète : deux domaines d'expertise qui sont souvent la prérogative de chercheurs appartenant à des communautés distinctes.

Plusieurs facteurs peuvent expliquer ce cheminement. Le plus évident est expliqué par la situation de la recherche publique italienne. Après mes études de doctorat, la réduction des financements publiques aux universités, et en conséquence la réduction de postes de chercheurs m'a poussé à plusieurs fois à prendre des postes temporaires dans plusieurs universités et groupes de recherche et, donc, à travailler sur des sujets les plus diverses. Évidemment, il ne s'agit que d'un facteur externe (d'une certaine fa**c**con, on pourrait dire le facteur déclencheur) pour interpréter mon parcours. En effet, avec le recul nécessaire, un privilège accordé par mon poste de travail actuel, il y a d'autres facteurs qui m'ont guidé sur cette voie, une voie qui aurait pu être différente, comme en témoignent les histoires et les choix d'autres chercheurs dans des conditions semblables aux miennes. Je crois que ma curiosité et mon désir de "résoudre les problèmes" ont permit de m'adapter à des nouveaux environnements, à étudier des nouvelles techniques d'optimisation et à les appliquer à des domaines d'application très variés.

Avant de présenter un grille de lecture du manuscrit, et quels thèmes j'ai choisi de présenter, je décris dans un premier temps, mon parcours en recherche. Depuis mes années d'études secondaire, j'ai toujours été passionnée par les mathématiques, et par conséquent par l'informatique, que j'ai toujours considérée comme une version (ou une évolution) appliquée des mathématiques. Ceci explique mon choix de poursuivre mes études universitaires en génie informatique, car je pensais que c'était la meilleure fa**c**con de continuer à étudier les mathématiques, tout en offrant de bonnes possibilités d'emploi une fois terminé mes études.

C'est au cours mon travail de fin d'études que j'ai commencé à travailler sur des méthodes d'optimisation globale appliquées à un problème de bio-informatique : le docking des protéines, c'est-à-dire la détermination de la configuration de stabilité (énergie minimale) que deux protéines assument quand elles sont en interaction étroite (docking - amarrage). Ce travail et sa présentation lors d'une conférence internationale, où des personnes dont je connaissais le nom pour avoir travaillé sur leurs livres, étaient assises tout

proche de moi, m'ont donné l'envie de commencer mes études doctoraux à l'Université de Florence pour continuer à travailler sur l'optimisation globale.

A partir du docking des protéines, le travail a évolué vers la recherche de méthodes d'optimisation globale pour les fonctions dites "funnel" (entonnoir), c'est-à-dire des fonctions qui ont un nombre très élevé de minimum locaux, mais avec une structure cachée plus simple (des fonctions convexes perturbées par un bruit). Avec un exemple visuel : en regardant de près on voit beaucoup de petites vallées très similaires, mais en se déplaccant "assez loin" une structure simple émerge en montrant une seule ample vallée. La raison de ce choix était qu'il y avait de bonnes raisons de croire que le docking des protéines partageait une partie de sa structure avec le problème de folding (repliement) des protéines (un sujet de bio-informatique très populaire à l'époque). La principale contribution de ma thèse a été de proposer des méthodes d'optimisation pour aborder l'optimisation des fonctions funnel. Les éléments clés des ces méthodes sont l'utilisation de l'optimisation locale standard (type descente du gradient) dans une procédure d'optimisation globale qui exploite la structure funnel du problème.

Un facteur central lorsqu'on travaille dans le domaine de l'optimisation globale est la difficulté de démontrer que la solution trouvée est réellement la meilleure solution globale, et pas seulement une solution locale (c'est-à-dire la meilleure en absolut). Bien qu'il existe des méthodes exactes d'optimisation globale, c'est-à-dire des méthodes qui peuvent non seulement apporter une solution, mais aussi certifier l'optimalité, dans certains applications/problématiques, ces méthodes ne peuvent pas être appliquées car elles nécessitent des hypothèses sur la structure du problème qui ne sont pas toujours satisfaites ou ne sont pas toujours capables de résoudre des problèmes à grande taille. Pour cette raison, une pratique courante de la communauté est de partager des bases de données sur les problèmes d'optimisation difficiles et avec les meilleures solutions trouvées jusqu'à présent, ce qu'on appelle les points d'optimum putatifs. Cela permet d'avoir un terrain commun pour comparer les méthodes de résolution et, il crée, comme effet secondaire, une sorte de "jeu" dans la communauté : puis-je trouver une meilleure solution que celles trouvé par d'autres chercheurs ? Cet esprit a certainement marqué mes recherches, notamment dans le choix des cas d'application.

Dans les premières années qui ont suivi ma thèse, j'ai travaillé sur quelques variantes de circle packing (arrangement compact des cercles dans un conteneur), améliorant certains des records existants et participant (et gagnant) à un concours international d'optimisation. C'est en présentant ce travail lors d'une conférence d'optimisation que j'ai découvert le problème de la conception optimale des trajectoires spatiales. Ce dans cette conférence que un chercheur (spécialisé dans l'aérospatiale) m'a dit : "Si tu es si doué pour résoudre des problèmes d'optimisation globale, pourquoi tu n'arrêtes pas de *jouer* avec tes cercles et tu essayes de résoudre nos problèmes ?" Et voilà, un nouveau défi : trouver des méthodes d'optimisation globales efficaces pour un problème d'optimisation back-box (boîte noire) très difficile, et aussi comprendre, au moins dans ses principaux aspects, une nouvelle application.

Les applications réelles sont donc un élément récurrent de mes recherches. Cela s'explique en partie par les besoins dictés par les différents contrats de recherche et/ou industriels sur lesquels j'ai travaillé. Cependant, je crois que mon attention pour les

problèmes d'optimisation d'origine applicative a également été motivé par mon intérêt pour les problèmes concrets et la nécessité de trouver un équilibre entre la théorie et la pratique (un élément toujours présent et certainement renforcé par ma formation ingénieur).

C'est dans cet esprit que, lorsque j'ai eu l'opportunité d'intégrer un projet au Politecnico di Milano en collaboration avec Alcatel-Lucent, j'ai décidé de "faire un saut" à la fois géographique et méthodologique. L'objectif principal du projet était de proposer des modèles de programmation mathématique et des algorithmes de résolution pour permettre à Alcatel-Lucent de répondre aux appels d'offres privés et publics en proposant des projets de réseaux de télécommunications rentables et performants. Donc, une autre application de l'optimisation avec son propre langage, à savoir les réseaux de télécommunications, et un autre domaine d'optimisation, l'optimisation discrète. Le travail au Politecnico s'est naturellement étendu de ce problème de conception de réseau à d'autres problèmes de Technologies de l'Information et de la Communication (TIC) : green-networking (planification des réseau à basse impact environnemental), optimisation des data-centers (centres de données), facility-location (problèmes d'emplacement d'installations). Ces thèmes constituent une part importante de mes recherches actuelles, avec des travaux allant de la modélisation aux algorithmes exacts et à les heuristiques.

Depuis décembre 2010, grâce à mon intégration dans un poste de chercheur associé financé par le Ministère italien de la Recherche, j'ai commencé à travailler sur l'optimisation des systèmes de santé. Bien que le sujet était significativement différent des précédents, j'aime penser que mon expérience en optimisation et dans des domaines appliqués m'aie permis d'apporter une contribution originale au projet, où j'ai principalement travaillé sur des modèles de programmation mathématiques pour la planification des interventions chirurgicales.

En septembre 2013, j'ai intégré l'Ecole Nationale Supérieur des Mines de Nancy et le LORIA (Laboratoire Lorrain d'Informatique et ses Applications) dans un poste de maître de conférences.

Je pense que ce récit puisse expliquer, du moins en partie, la difficulté de résumer mes recherches de manière concise et, surtout, cohérente. En fait, mes contributions de recherche couvrent différentes approches méthodologiques (des heuristiques aux méthodes exactes, en passant par certaines contributions sur la complexité computationnelle), différentes "branches" d'optimisation (de l'optimisation non linéaire non convexe à l'optimisation discrète) et différents domaines d'application. Cependant, comme pour les fonctions funnel, en regardant mon travail à distance, une structure commune émerge : une prédilection pour les problèmes d'optimisation découlant des applications, un travail d'abstraction visant à mettre en évidence la structure sous-jacente de ces problèmes dans le but d'identifier les similitudes avec des problèmes connus dans la littérature ou, inversement, d'établir leur caractère innovante dans le but ultime de proposer des méthodes efficaces pour résoudre le problème applicatives de départ.

Ce sont ces éléments que j'ai essayé de faire resortir avec la sélection des mes travaux pour la rédaction de ce manuscrit. L'autre critère était de répondre à la nécessité de fournir tous les éléments nécessaires pour introduire et expliquer les perspectives de recherche. Pour ces raisons, mes contributions (et aussi mon projet de recherche) sont

divisées en deux axes : l'un consacré à l'optimisation globale et l'autre aux problèmes d'optimisation discrète pour les applications en TIC, qu'elles, à mon avis, représentent les deux principales catégories qui peuvent être utilisées pour classifier la plupart de mes contributions. Chaque partie se termine par des travaux récents directement liés à mon programme de recherche.

En ce qui concerne l'optimisation globale, j'ai décidé de mettre l'accent sur les travaux qui portent sur les algorithmes métaheuristiques. Pour cela, d'abord j'introduis des idées générales d'optimisation globale et quelques méthodes métaheuristiques simples, puis j'introduis le concept d'intégration de l'information (locale et globale) dans les méthodes ci-dessus énoncés afin de résoudre des problèmes d'optimisation difficiles. Les fonctions funnel sont un bon exemple : une phase de smoothing (lissage) peut être ajoutée pour éliminer les oscillations et faire ressortir la structure sous-jacente. Mes travaux sur le circle packing et la conception de trajectoires spatiales sont utilisés comme exemples d'application de ce principe à différents problèmes d'optimisation globale. Dans les deux cas, les méthodes proposées ont permis d'identifier des nouvelles solutions par rapport à celles de la littérature (optimum putatifs). Je conclus la discussion sur les problèmes d'optimisation globale introduisant un de mes travaux les plus récents : l'optimisation des procédés de séparation de gaz par des membranes. Les membranes sont des dispositifs industriels qui, par construction, permettent de séparer, grâce à l'application d'une différence de pression, un mélange gazeux en ses composants, permettant par exemple d'extraire l'azote de l'air, ou de purifier le méthane. Pour atteindre un niveau de qualité satisfaisant en réduisant les coûts du système, il faut interconnecter davantage plusieures membranes et établir des conditions opérationnelles appropriées pour le système. Le problème d'optimisation résultant est un problème de programmation non-linéaire en variables mixte (continues et entières) à grande échelle qui, même en négligeant les variables entières, s'avère être un problème d'optimisation globale très difficile à résoudre. Notre contribution sur le sujet est d'avoir proposé un algorithme de résolution basé sur l'optimisation globale et d'avoir prouvé son efficacité à la fois sur une étude de cas de la littérature et sur une étude de cas de dérivation industrielle. Cette application, et le défi de l'optimisation associées, sont le point de départ d'un des deux axes des mes perspectives de recherche.

En ce qui concerne l'optimisation discrète des applications TIC, j'ai décidé de me concentrer sur une sélection d'œuvres qui marquent à la fois un parcours temporel et thématique.

Tout d'abord, je présente mes principales contributions sur la gestion et la planification des réseaux de télécommunications à faible impact environnemental (green-networking). Le texte se développe autour de la gestion des réseaux de télécommunication pour les réseaux IP, c'est-à-dire le routage et l'alternance marche/arrêt dynamique des dispositifs afin de permettre une réduction de la consommation énergétique sans dégrader le niveau de qualité de service (QoS). À ce sujet, je présente un certain nombre de contributions : tout d'abord, la rédaction d'un état de l'art, dans une perspective d'optimisation et de programmation mathématique, qui permet de montrer quels modèles peuvent être considérés comme innovants par rapport aux problèmes le plus classique de la conception de réseaux (network design), et lesquels sont essentiellement des modèles connus ;

deuxièmement, une analyse de l'impact des différentes stratégies de protection est présentée en termes de consommation d'énergie et de congestion ; enfin, inspirée par ce problème appliqué et motivée par la nécessité de trouver des algorithmes de résolution plus efficaces pour certaines stratégies de protection, un nouvel algorithme de résolution basé sur une re-formulation par projection est proposé et son efficacité démontrée computationnellement.

Ensuite, deux contributions sur des problématiques émergentes dans la convergence des réseaux de télécommunication et des centres de données sont présentes. Cette convergence représente une nouvelle perspective pour la communauté des TIC, où historiquement la communauté des réseaux de télécommunication et la communauté des centres de données/cloud sont indépendantes. Les nouveaux problèmes d'optimisation qui émergent de cette perspective et le défi que représente leur résolution sont l'autre axe autour duquel mon projet de recherche se développe.

Sur ce point, la première contribution concerne la gestion conjointe des centres de données et des réseaux de télécommunication pour permettre une exploitation optimale des énergies renouvelables. À l'aide d'un modèle de programmation mathématique, l'impact de la gestion partagée des ressources du réseau et du système est analysé. La possibilité de déplacer intelligemment les demandes entre centre de donnée à l'aide du réseaux permet d'utiliser au mieux les ressources renouvelables (solaires, éoliennes, géothermiques) qui, par leur nature, sont volatiles et, donc, doivent être utilisées quand (et où) elles sont disponibles. Notre modèle, et l'analyse des résultats obtenus avec celui là, montrent clairement les avantages d'une telle gestion partagée, qui montre comme soit possible et envisageable une adoption plus importante des ressources renouvelables par les plus grands fournisseurs du cloud.

Cette application peut être considérée comme le "trait d'union" entre mon travail précédent sur les green-networking (avec quelques contributions sur la gestion des centres de données) et mon travail plus récent sur l'emplacement et le routage pour Virtual Network Functions (VNF), ou fonctions de réseau virtuelles. Notre contribution à ce problème couvre deux aspects différents : premièrement, on a proposé des modèles de programmation mathématique et une analyse des les possibles solutions sur une étude de cas réaliste, deuxièmement, on a fourni une analyse des propriétés du problème en termes de complexité et une étude du comportement des formulations mathématiques de programmation existantes dans la littérature, tant en terme de propriétés théoriques que en terme de résultats computationnelles.

Je pense que ces contributions represents bien la structure "cachée" de la quel j'ai parlé auparavant. En effet, dans plusieurs travaux des nouveau modèles de programmation mathématiques ont été proposés afin de résoudre des problématiques appliquées (comme pour le green-networking, le management conjoint de réseaux de télécommunication et centres des données, le placement et routage de VNF). De plus, plusieurs contributions visent à mettre en évidence et analyser la structure mathématique sous jacent et les propriétés des problèmes étudiés (l'état de l'art sur le green-networking pour souligner les similarités et les différences entre le network design "classique" et le problèmes des green-networking, la comparaisons entre différentes formulations pour le problème de VNF, la représentation "multi-échelle" des variables dans le procédés de séparations gazeuse

par membranes, etc.). Enfin, l'information obtenue dans cette analyse est utilisé pour la conception des algorithmes d'optimisation performants qu'ils ont permis d'améliorer l'état de l'art (la formulations projeté pour les problèmes de protection dans le réseaux, la structure funnel pour le circle packing et la conception optimale des trajectoires spatiales, etc.)

Comme j'ai anticipé, mon projet de recherche est organisé autour de 2 thèmes principaux par rapport aux applications : un est centré sur l'optimisation des problèmes émergents dans l'intégration de réseaux de télécommunications et centres de données, l'autre sur les méthodes d'optimisation pour la génie des procédés (dont les procédés de séparations gazeuse par membranes sont un cas particulier). Dans un point de vue méthodologique, le premier thème porte sur la conception des méthodes de résolution exacts et heuristiques pour des problèmes de programmation à nombre entières issues de la combinaison de facility location et network design, le deuxième thème porte sur les méthodes d'optimisation globale et programmation non-linéaire en variables mixtes avec l'accent sur les méthodes heuristiques. Le dénominateur commun des ces deux thèmes est l'étude des problèmes complexes issus de milieux industrielles. D'un point de vu plus méthodologique, le point focal serait l'analyse des propriétés structurelles des problèmes et la conséquente possibilité d'utiliser des méthodes de décomposition pour porter à la surface des structures connues et l'exploitation des telles informations dans la conception d'algorithmes de résolution efficaces pour des problèmes à grande échelle.

Contents

1 Introduction

This manuscript is the result of my attempt to summarize my research journey and to illustrate my research prospects for the near future. This is a difficult task mainly for two specific reasons: first of all, I had the wish to write a document that could be accessible by the large audience of computer scientists (still assuming some basic knowledge on optimization), second, although I have always worked in the field of optimization, my path can be considered rather atypical. In fact, in a relatively short period of time, I have moved from global (non-linear, non-convex) optimization to discrete optimization: two field of expertise that are often prerogative of researchers belonging to separate communities.

To explain the reasons for this path, it is necessary to take into account several factors. The most evident is certainly the Italian public research situation after my PhD studies: the reduction of fundings and the lack of stable positions led me to move to different universities/research groups and, therefore, to work on very different topics. Obviously, although this is one of the key factors to interpret my path, it is only the external one (in some sense, the trigger). In fact, looking back in perspective, a privilege given by my current stable position, there are other factors that have guided me along this path, path that could have been different, as evidenced by the history and the choices of others researchers in similar environmental conditions. I believe that my natural curiosity and my wish "to solve problems" helped me to adapt to new environments, to study new optimization techniques and apply them to very different applications domains[1].

Before explaining how I decided to organize the manuscript and which material I selected, I want to take a step back to briefly "tell the story of my research path". Since I was a child, I have always been passionate about mathematics and, for extension, about computer science which I have always seen as an "applied version" of mathematics itself. This explains my choice to continue my university studies in Computer Science Engineering, which I felt was the best way to continue studying mathematics, while providing good job opportunities.

It was during my master thesis that I started to work on global optimization methods applied to a bio-informatics problem: protein docking, i.e. establish the configuration of stability (minimum energy) that two proteins assume when they are in strict interaction (the docking). The master thesis' work and presenting it in an international conference where people I knew by name for having studied from their books were sitting two benches away from me[2] gave me wish to enroll in the PhD program of the University of Florence to continue working on global optimization.

[1] A little stubbornness typical of my family completes the picture.

[2] the fact that the conference was in the lovely Erice is, probably, an additional influential element

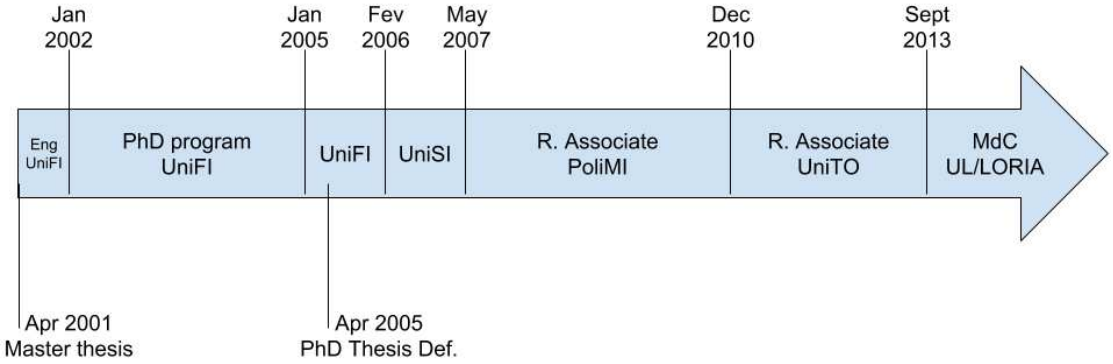


Figure 1.1: Time-line from master thesis defense up to now

Starting from protein docking, the work evolved on the search of global optimization methods for the so called "funnel functions", i.e. functions that exhibit a very high number of local minima, but with a hidden simpler structure (like a convex function perturbed by a noise). With a visual example: looking closer a lot of similar small valleys exists, but moving "far 'enough" the underlying simple structure emerges showing a single valley. The reason behind this choice was that there were good reasons to believe that protein docking, as the closer related protein folding (a very hot topic of bio-informatics at the time), exhibited such a structure. The main contribution of my thesis was to propose optimization methods to tackle the optimization of funnel functions. The key elements of such methods are the use of standard local optimization (gradient-descent like) inside a global optimization procedure that exploits the funnel structure of the problem.

A key issue when working on global optimization is the difficulty of proving that the found solution is actually the global optimum (best overall). Although there exist exact methods for global optimization, i.e. capable not only to provide a solution, but also to certify its optimality, they cannot always be applied because they require hypotheses on the problem structure that are not always satisfied or they are not suitable for solving large size problems. For this reason, a common practice is to keep databases with difficult global optimization problems and the best solutions found so far, the so-called putative optima. This allows to have a ground for comparing different solution methods and, as a side effect, creates a sort of "game" in the community: can I find a better solution than the others? This spirit has certainly marked my research, especially in the choice of the application cases.

In the first years after the PhD thesis, I worked on some variants of circle packing, improving some of the existing records and participating to (and winning) an international optimization competition. It was presenting such a work at a global optimization conference, that I discovered the problem of optimal spatial trajectory design. In fact, an aerospace researcher, told me "if you are so good to solve global optimization problems, why don't you stop playing with circles and try to solve our problems?". There started a new challenge: to find effective global optimization strategies for a challenging

black-box optimization problem, but also to understand, at least in its main aspects, a real application.

Indeed, real applications are a recurrent element of my research. It is partly explained by the needs dictated by the various research and/or industrial contracts I worked on. However, I believe that the focus on optimization problems arising from applications was also motivated by my interest in concrete problems and the need to find a balance between theory and practice (element always present and certainly reinforced by my engineering studies).

It is in this spirit that when I had the opportunity to integrate a project at the Politecnico di Milano in collaboration with Alcatel-Lucent[3], I decided to move both geographically and methodologically. The heart of the project was to propose mathematical programming models and resolution algorithms to allow Alcatel-Lucent to determine optimized telecommunication network designs for responding to private and public calls. So, another application problem with its own language, telecommunication networks, and another area of optimization, discrete optimization. The work at Politecnico, from this network design problem, naturally extended to other Information and Communication Technology (ICT) problems: green-networking, data-centers optimization, facility location. These themes form an important portion of my current research, with works ranging from modeling to exact and heuristic algorithms.

Since December 2010, thanks to my integration in a position of research associate funded by the Italian Research Ministry, I started working on health-care systems optimization. Even if this topic was significantly different from the previous ones, I like to believe that my past experience both in optimization and application problems helped me to give an original contribution to the project team work, working mainly on mathematical programming models for surgical planning.

Finally, in September 2013, I integrated the Ecole de Mines de Nancy and the LORIA (Lorraine Laboratory in Computer Science and its applications) in a position of Maître de Conferences (Associate professor).

I believe that this account justify, at least in part, the difficulty to summarize my research in a concise and, above all, cohesive manner. Indeed, my research contributions cover different methodological approaches (from heuristic to exact methods, passing through some contributions on computational complexity), different "branches" of optimization (from non-convex non-linear to discrete optimization) and several different application fields. However, as with funnel functions, looking to my work from a distance, a common structure emerges: a predilection for optimization problems arising from applications, a work of abstraction to highlight the underlying structure of such problems with the aim of identifying similarities with problems known in the literature or, on the contrary, to establish their novelty, with the ultimate aim of proposing effective resolution methods for the original application problem.

These are the elements that I tried to highlight with my works' selection. The second criterion I used was dictated by the need to introduce the necessary elements to explain at best my research perspectives. For this reason, my contributions (as my research

[3]Just before a short parenthesis were I was working at the University of Siena.

project) are divided in two parts: one dedicated to global optimization and a second one to discrete optimization for ICT applications, that I believe are the main two categories that can be used to classify the majority of my works. Each part is concluded by some very recent contribution that directly connects to my research perspectives.

For what concerns global optimization, I decided to focus on my research work on meta-heuristic algorithms. To this aim, first I introduce some general ideas about global optimization and some basic meta-heuristics, then I introduce the concept of integrating (global and local) information in a simple meta-heuristic to allow solving some challenging problems. Funnel functions are a good example: a smoothing phase can be added to remove oscillations and let emerge the underlying structure. My contributions on circle packing and space trajectory design are reported to explain how this concept can be applied to different problems. For both of problems, some new putative optima were found using the prosed optimization methods.

The discussion on global optimization is concluded by one recent work: the optimization of membrane gas separation processes. Membranes are industrial objects that by construction, if an opportune difference of pressure is applied, allow to separate a gas mixture in its components, for example extracting nitrogen from air, or purifying natural gas. To obtain the necessary performances at a competitive costs, multiple membranes must be interconnected in a complex system and opportune operational conditions must be selected. The resulting optimization problem is a large Mixed Integer Non-Linear Problem (MINLP), which, even fixing the integer variables, results in a challenging global optimization problem. Our contribution on the topic is to have proposed a global optimization algorithm and prove its efficacy on a case study from the literature and an industrial case study. This topic and the optimization challenge associated to it are the starting point of one of the two research lines of my project.

For what concerns discrete optimization for ICT applications, I decided to focus on a selection of works to mark a path that is temporal, but also thematic.

First, the main contributions on green networking are summarized. The focus is on energy-aware network management for IP-networks, i.e. the routing and the dynamical switching on/off of the devices to allow energy consumption reduction without affecting the requested QoS. On this problem, the contributions are of different nature: first of all, a comprehensive literature review with the focus on mathematical programming models is proposed, allowing to show where the models existing in the literature have introduced some novelty with respect to "classical" network design (and management), and when they are just a reinterpretation in the "energy-aware perspective" of well known models; second, an analysis on the impact of different protection strategies is reported in term of energy consumption and congestion; last, but not least, inspired by this application problem, and driven by the need of more effective algorithms for solving some specific protection models, a new solution algorithm based on a projected formulation is proposed and its effectiveness was showed.

Second, two contributions related to problems arising from the convergence of network and computing systems management are presented. The reason of this choice is motivated by the fact that this is a quite new perspective in ICT, where historically the telecommunication and the computing systems community have been separated. The

new optimization problems resulting from this convergence and the challenge related to their solution is the other axis around which my research project is developed. On this topic, the first contribution is on the joint management of service centers and communication networks to exploit green-energy resources. A mathematical programming model is proposed to analyze the impact of managing jointly multiple services centers, allowing the use of the network to serve demands originated in one service center in another one. The possibility to move "smartly" demands allows to fully exploit green-energy sources (as solar, wind or geothermic), that for they nature are volatile, and therefore must be used when (and where) available. Our model, and the analysis of the results we obtained using it, shows clearly the advantages of such a joint management advocating a larger adoption of green energy sources for the largest Cloud provides.

This application can be considered as the "trait d'union" between my previous works on green network design and management (with some contributions also on data-centers management) and the most recent works on the Virtual Network Functions (VNF) placement and routing. In this second application, the resulting optimization problem is a novel combination of a network design problem and a facility location one. Our contribution to this problem is twofold: first, we provided mathematical programming models and a deep analysis on the obtained results for a realistic problem setting (taking into account among other features: different latency regimes, compression/decompression of demand bandwidth along the routing[4]), second, but not less important, we provided an analysis of the properties of the problem in terms of computational complexity and we compared, theoretically and numerically, the current state of the art formulations.

I believe that these contributions represent well the hidden structure I mentioned above. Indeed, in several of these works new mathematical programming models were proposed to tackle different applications (such as the green-networking for IP-networks, the joint management of network and service centers, the VNF placement and routing). Furthermore, several contributions aim at highlighting and analyzing the mathematical structure and the resulting properties of the problems under analysis (the survey on green networking to highlight the similarities and differences between network design and green networking problems, the comparison between different formulations for the VNF problem, the "multi-scale" presentation of variables in membrane gas separation systems, etc.). Finally, the obtained information is used to design algorithms to improve upon the current state of the art solutions (the projected formulation for the shared protection, the funnel structure for circle packing and space design trajectories, etc).

My research project is articulated in two main lines from the application point of view: one dedicated to the optimization problems arising from the integration of network and computing systems, the other on optimization methods for process synthesis problems (of which the membrane gas separation processes represent a specific case). From the methodological point of view, the first line will be centered on exact and heuristic methods for integer programming problems resulting from the combination of facility location and network design, the second line will be centered on global optimization and MINLP methods with the main focus on heuristic methods. The common denominator of the

[4]This contribution is just shortly discussed in the manuscript due to space reasons

two research lines will be the study on complex problems arising from real applications. From the methodological point of view, the focus will be on the analysis of the problems' features, the possibility to use decomposition to expose known structures and to exploit such information to develop efficient algorithms for solving large scale problems.

1.1 Manuscript organization

To allow the reader to easy identify the references to my contributions (and their nature), I use coding letters: J for journals, C for conferences with review, B for book chapters and finally O for all the others. If no letter appears, I am not coauthor of the contribution. All my contributions are grouped at the end of the manuscript, all the other references are grouped after Chapter 6.

The material is organized along two axes: global optimization and discrete optimization for ICT problems. Two chapters are dedicated to each axis, the second one (in both cases) presents more recent results that directly connect to my research perspectives. I tried to introduce notation and informations in each chapter to allow reading them quite independently (I keep this in mind also in the reduction of the research perspectives). This partially explains some repetitions (and the residual presence of some inconsistency in terms of notation).

The first two chapters are dedicated to optimization problems:

- in Chapter 2 after a brief introduction on global optimization (with some examples extracted from [O9]), I report on contributions on circle packing and space trajectory design ([J21], [J20], [J19]);
- in Chapter 3 the membrane gas separation problem is introduced and our first results summarized ([O1], [J1]).

The next two chapters are dedicated to discrete optimizations problems in ICT:

- in Chapter 4, the main contributions on energy-aware network management are summarized ([J12], [J11], [J7], [J4])
- in Chapter 5 contributions on two problems arising from the convergence of telecommunication networks and computing systems are reported, namely green joint management of network and data-centers ([J13]) and VNF placement and routing ([C2],[J2], [J3], [W1])

My research perspectives are presented in Chapter 6. The manuscript is completed by a detailed curriculum vitae (Chapter 7), a short summary of my research activity (Chapter 8) and the full list of my publications (Chapter 9).

To complete this part, I report a synthetic view of my contributions.

Global optimization

Works contained in my PhD thesis ([O9]):

- protein docking [J27]
- general methods for "funnel structured" problems [J26],[J25]

Global optimization methods (heuristics) applied to different applications:

- Circle packing [J21], [J20]
- Space trajectory design [J19], [J10]
- Membrane gas separation processes: [O1] (this work was developed during the master project of Magdalena Krzaczkowska), [J1] (this work was developed during the ongoing PhD studies of Marjan Bozorg)

other contributions:

- test functions for global optimization [J24]
- theoretical bounds for inter-atomic distance in molecular clusters [J23], [J18].

Discrete Optimization for ICT applications

A significant part of the work on discrete optimization is centered on

- energy-aware management in communication networks [C7], [C6], [C5], [C4], [J12], [J11], [J7], [J5], [J4]. These works (except the last two) were developed during the PhD studies of Luca Giovanni Gianoli.

Others ICT energy-aware problems:

- joint management of communication networks and service centers with the aim of optimally utilize green energy sources [J13];
- radio-networks management. This work is not yet published, but the methods and results are described in two master reports (M. Raco and G. La Rosa) and presented in two conferences [O3, O2];
- energy-aware sensors networks [C1]. This work was developed during the PhD studies of Evangelia Tsionstiou.

Slightly moving from the energy-aware subject, I worked on other ICT optimization problems. Some are related on service center optimization:

- service centers management with QoS constraints [C9],[J16]. Part of this work was developed during the master project of Fosco Angelo Bombardieri.

All of the others are the direct result of the collaboration with Alcatel-Lucent:

- a network design problem for two-layer networks [J22];
- network loading with SRG-based protection [O5, O6].

or they inspired by it:

- a two-level facility location problem arising from the Fiber-to-the-home problem [J17, J15];

From my arrival in France, I started to work on another telecommunication network problem:

- the Virtual Network Functions placement and demand routing:
  - [C2], developed during the PhD studies of Dalal Belabed
  - [J2], [J3], [W1], developed during the (ongoing) PhD studies of Meihui Gao.

From a methodological point of view, these contributions can be synthetically organized in the following way (I consider the main contribution as some papers can fit in more than one category):

- innovative mathematical programming models: [J22], [C9], [C7], [C6], [C5], [C4], [J12], [J11], [J13], [C2], [C1], [J2]
- survey work: [J7];
- heuristic methods: [J17], [J16], [O2]
- exact optimization methods: [O3], [J15], [J5], [J4]
- computationally complextity, formulation performances and properties: [O5, O6], [J2], [W1]

Other contributions

- Critical Node Problem: [J14], [J8]
- Health-care problems: [O7], [O8], [B1], [B2], [O4], [C3], [J9], [J6].

2 Some contributions on Global optimization

Global optimization, even if largely applied also outside the optimization community, or maybe for this very reason, is a branch of optimization quite new and less known even by optimization experts and practitioners. Therefore, I decided to add a short section (Section 2.1) to introduce some general concepts and the notation I need to present my contributions in a more organic way.

The expert reader can skip this part, and must be advised that if she/he decides differently, she/he will find inaccuracies and missing parts due to the effort of make this part accessible and short. In fact, this section is not meant to be a survey on global optimization, neither a short version of it (the interested reader can refer to [110] for a quite comprehensive overview), but it has been thought as a "story-telling", in the sense that I describe some few concepts and methods (that are the building blocks used in my research) and explain the reasoning behind them. As a consequence, my description is biased by my own research experience.

2.1 A very short tour on global optimization

We will focus on continuous optimization:

$$P \begin{cases} \min f(x) \\ x \in S \subset \mathbb{R}^n \end{cases}$$

where $f : S \subset \mathbb{R}^n \Leftarrow \mathbb{R}$ is a sufficiently smooth function and $S \subset \mathbb{R}^n$ is a compact set. Under these hypotheses the **existence** of the minimum value $f^\star = f(x^\star)$ is guaranteed, where

$$x^\star \in S : f(x^\star) \leq f(x) \quad \forall x \in S$$

represents the/a corresponding global optimum point[1].

The problem that remains open is to determine **where** the global optimum is located and/or **which value** it assumes. For many years, global optimization as a discipline was put aside, because it was considered too difficult and, even worst, ill-defined. In fact, in 1978, Dixon proved that problem P is inherently unsolvable in a finite number of steps.

[1] multiple points can correspond to the same objective value, in this case we talk about equivalent solutions

Just to give the proof's flavor (that can be found in Dixon's paper [62]), let us consider a box-constrained optimization problem:

$$P \begin{cases} \min f(x) \\ x \in [a,b]^n \end{cases}$$

Suppose that we have sampled a given number of points in the feasible region, and evaluated the objective function f on all of them (see Figure 2.1a). Using this information, we can infer a "given shape" for f and, therefore, the position of the global optimum point (indicated by a full dot in Figure 2.1b). But, the collection of a new sample (as the one shown in Figure 2.1c) can prove that our guess was completely wrong. A new "guess" on the shape of f can be made as shown by the smooth line in Figure 2.1d. Then, a new sample can be found such that the procedure can be repeated again and again.

A little more formally, given a finite number of sample points, no matter how many, an infinite number of functions can interpolate such data, even evaluating in their correspondence not only the objective function f, but also its n-order derivatives (f, f', f'', ... f^n). Therefore, it is impossible to determine an unique "shape" for the objective function, and, as a consequence, a candidate point to be the global optimum.

Some additional information is needed to have a well-define global optimization problem. For example, if we can assume that problem P is a convex problem, i.e. f and S are convex, than any local optimum corresponds also to a global optimum, and the problem becomes solvable, at least numerically (see [33],[135] for comprehensive guides on convex optimization). In a very simplified way, such global information (convexity) allows to determine bounds on the behaviour of f "among two samples"[2], and therefore to concentrate the search in the region where "forcefully" the optimum is located. To come back to our example, if f is convex, then the new sample added in Figure 2.1c cannot exist, and the global optimum can be searched in the interval $[5, 7]$.

Convexity is a very strong assumption and it does not hold for many interesting optimization problems, nevertheless, there exists other forms of global information that allow to determine an (exact) solution method. Just to give some examples, we mention: quadratic programming[3], Lipschitz-continuous functions on a simple domain (box, simplex), low order polynomial functions, difference of convex problems, etc.

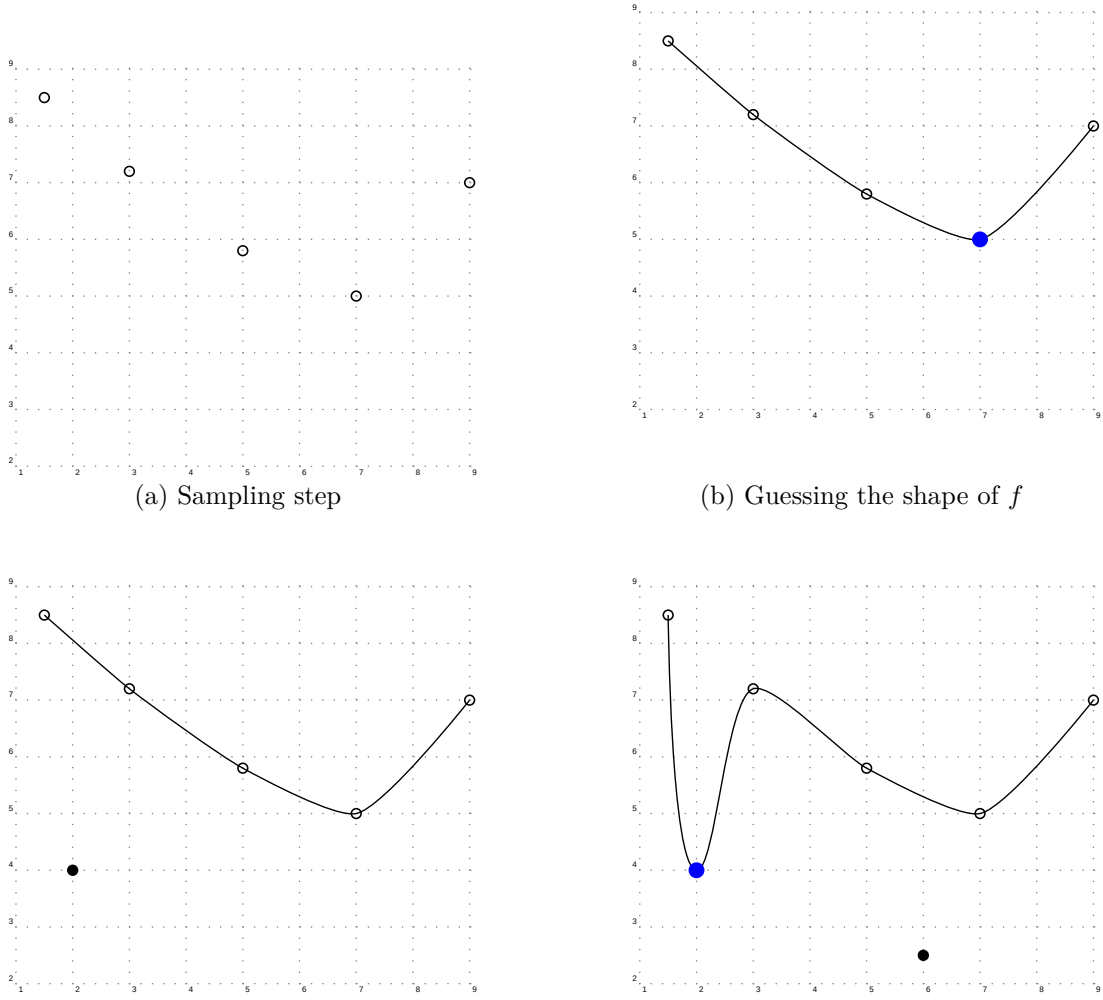
Even if weaker than convexity, these assumptions are not always valid, therefore the question remains open whether global optimization methods can be developed and applied to problems that do not present any information of these types.

Before trying to answer this question, two other questions must be answered: what does it mean exactly solving a non-linear (global) optimization problem and which kind of problems we are interested in.

To answer the first one, let us focus on numerical solutions and consider to accept solutions under a certain precision ϵ. In this case, we can consider that solving problem

[2] f is convex iff given two points x_1, x_2 any convex combination of them $y = \alpha x_1 + (1-\alpha)x_2$ with $\alpha \in [0,1]$ is such that $f(y) \leq \alpha f(x_1) + (1-\alpha)f(x_2)$

[3] Any concave quadratic function on a polyhedron assumes its minimum on a vertex.



(a) Sampling step

(b) Guessing the shape of f

(c) Sampling a new point

(d) A new guess for the shape of f, and ... a new sample

Figure 2.1: Looking for the global optimum

P means finding a point belonging to set:

$$S_\epsilon(f^\star) = \{x \in S : f(x) \leq f^\star + \epsilon\} \tag{2.1}$$

(see for example Figure 2.2) or, if we are more concerned about the distance from the "exact" global optimum, than about its value, in set:

$$B_\epsilon(x^\star) = \{x \in S : d(x^\star, S) \leq \epsilon\} \tag{2.2}$$

where $d(x, A)$ is a suitable distance between a point x and a set A. We need to observe

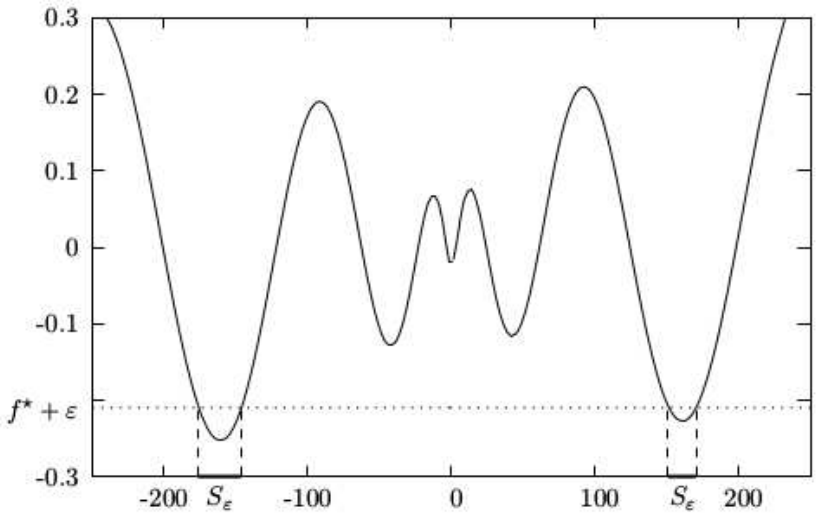
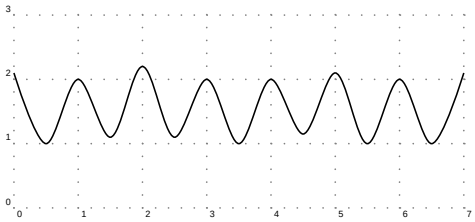


Figure 2.2: Numerical global optimum points

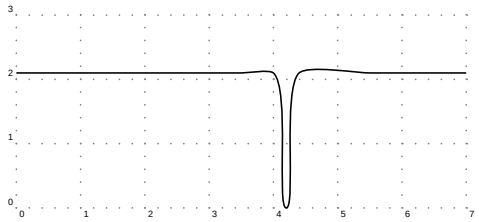
that, even finding a point in S_ϵ or B_ϵ is still a difficult problem (and ill-defined without any additional information).

To answer the second question, let us consider the three examples reported in Figure 2.3. The first two cases can be considered difficult to solve, but not interesting in practice. If all local optima are almost the same in value (as shown in Figure 2.3a), they can be numerically equivalent under the accepted numerical precision (or the precision needed by the application), therefore finding just a single local optimum can be enough. An almost flat objective with a single "hole" (as shown in Figure 2.3b) can produce an almost unsolvable problem (at least in probability). But, even if the global optimum is found, it would be "not useful" in practice, in fact a tiny change on the coordinates of such global optimum point could lead a significantly different objective value, and therefore with finite numerical precision it would be very difficult to use.

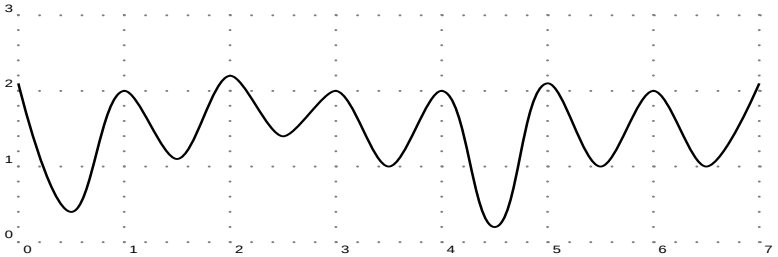
We are then left with the family of problems where one or some of the local optima are "significantly" different from the others in terms of value and/or position. For these problems, methods based on random sampling can guarantee a convergence in probability. If any additional information (global or local) is available, then basic strategies can be enriched to obtain better performances. In the following, I introduce some very basic meta-heuristics and I try to explain by examples what I mean with the terms local and global information and how this information can be used to modify such algorithms.



(a) Many "almost" equivalent optima



(b) Golf hole on an otherwise flat region



(c) Some local optima are "significantly" different from the others

Figure 2.3: Examples of global optimization problems

2.1.1 Some very simple meta-heuristic methods

The simplest *stochastic*[4] algorithm is known as Pure Random Search (PRS). At each iteration a random sample is selected[5] in the feasible region and it is illustrated in Algorithm1. Function GloballyGenerate() usually correspond to uniform generation in S. A classical stopping rule is given by a maximum number of iteration (N), whereas different rules, like the ones based on "apparent" convergence of the algorithm (no more improvement points found after a given number of iterations), can also be used.

Algorithm 1 `Pure Random Search`

procedure PURE RANDOM SEARCH(N)
  $x^\star$ =GloballyGenerate()
  while $n < N$ **do**
    z=GloballyGenerate()
    if $f(z) < f(x^\star)$ **then**
      $x^\star = z$
  return $x^\star, f(x^*)$

[4]Traditionally global optimization methods are divided in *deterministic* and *stochastic*, to the second group belong all methods that use some form of randomization in their decision process.

[5]Depending on the shape of feasible region, even uniformly random sampling can be a difficult task. Some possible ideas of how to perform this tasks are reported in the following for some applications.

The probability of finding the optimum is given by $\frac{V_{S_\epsilon}}{V_S}$, where V_S is the volume of the set S and V_{S_ϵ} is the volum of set S_ϵ (see Figure 2.2).

Using additional information on the problem to modify this simple algorithm can allow to improve the convergence probability. Such information can be divided in two main large classes: global information and local information. Let us consider two examples.

Local information Any global minimum is also a local minimum, therefore we can consider to use a standard local optimizer (gradient-descent, conjugate gradient, Quasi-Newton, etc) in conjunction with the PRS, getting the so called Multistart method (see Algorithm 2).

Algorithm 2 `Multistart`

```
1: procedure MULTISTART(N)
2:     x =GloballyGenerate()
3:     x* = locmin(x)
4:     while n < N do
5:         z=GloballyGenerate()
6:         z = locmin(z)
7:         if f(z) < f(x*) then
8:             x* = z
       return x*, f(x*)
```

The probability of finding the optimum using Multistart is $\frac{A_\epsilon}{V_S}$, where A_ϵ is the volume of the "basin of attraction" of points in $S_\epsilon^\star$ (see Figure 2.4). It is not obvious how to define a basin of attraction; the "common sense" is that a point x is in the basin of attraction of a local minimum point $\bar{x}$, if we can move from x to $\bar{x}$ "going down" along the objective function, (see [O9] for a tentative formal definition and some observations). We can observe that passing from PRS to Multistart is done introducing an *intensification* step in the search (made through a standard local search), i.e. a step that, instead of global exploring the overall space, tries to improve "locally" the current solution.

If we consider an ideal local optimizer, we can define function

$$\mathcal{L}_f(x) = \begin{cases} locmin_y(x) & f(y) \\ y \in S \end{cases}$$

where function $locmin(x)$ represent the solution obtained by a local minimizer using x as starting point. We can observe that minimizing $\mathcal{L}_f$ is equivalent to minimize f, therefore the Multistart algorithm can be imagined as a PRS applied to $\mathcal{L}_f$ instead to f. Many stochastic global optimization algorithms can be modified accordingly, considering to work on the ideal function $\mathcal{L}_f$ instead of f, simply adding a standard local search step after each sampling. We observe that, from a practical point of view, $\mathcal{L}_f$ is implicitly defined and **in practice** depends on the

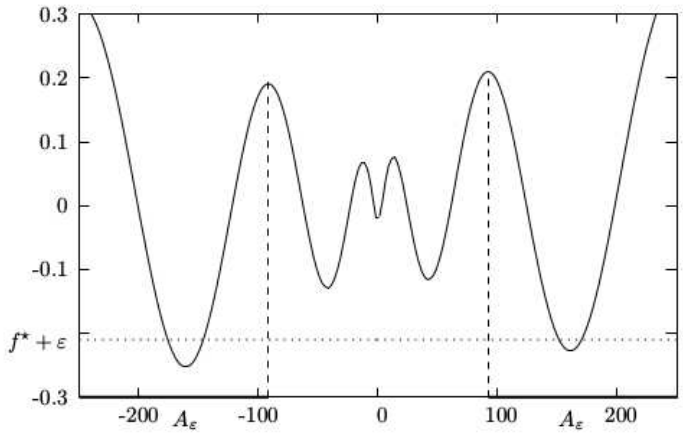


Figure 2.4: Multistart success probability

local minimizer used. In particular, if f is not locally convex, the local minimum can belong to a different basin of attraction with respect to the starting point x, or even points that are not local minima can be found, for example, any KKT point ([135]).

Global information In many global optimization problem, the local optima are not randomly displaced in the feasible space, but follow some sort of ordered structure. For molecular conformations problem (from protein folding to atomic clusters), the objective function shows the so called *funnel structure* ([104]), i.e. the objective function can be imagined as the sum of a "simple" one (with a single or few local minima) and a "noise" that introduces a high number of local minima (see Figure 2.5 for an example and [169, 109] for possible definitions).

In this case, it would be more effective, instead of sampling completely at random, to sample in the *neighborhood* of the current point, introducing a new *intensification* phase, but at a different "level", instead of considering continuous local optimization, taking into account adjacent local optima, i.e. a sort of combinatorial structure between them (see [109] for a discussion on the structure of global optimization problems and the difficulty associated to them). The resulting algorithm that we will call, for similarity with a strategy used in combinatorial optimization, Iterated Local Search (ILS) is stretched in Algorithm3: after generating an initial starting point on the overall search space, new points are generated "near" this one, using an appropriate LocallyGenerate() function. When one of such new points improves upon the current "center", it substitutes it (see lines 5-6 of Algorithm 3), and the search continues in a new neighborhood.

A common StoppingRule is waiting until a maximum number of consecutive *unsuccessful* iterations are performed, where unsuccessful means that the new generated

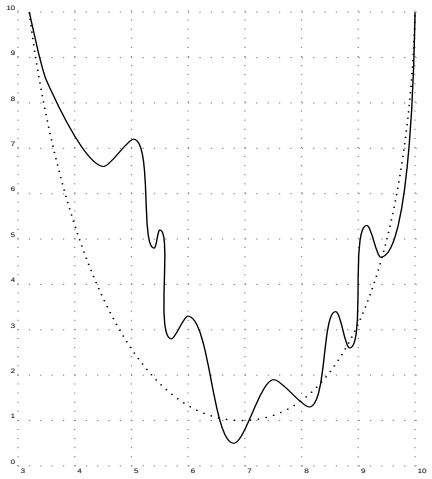


Figure 2.5: An example of funnel function

Algorithm 3 `Iterated Local Search`

```
1: procedure ITERATED LOCAL SEARCH
2:     x* =GloballyGenerate()
3:     while StoppingRule is false do
4:         z=LocallyGenerate(x*)
5:         if f(z) < f(x*) then
6:             x* = z
       return x*, f(x*)
```

point does not improve with respect to the current record. This stopping criteria allows to continue the search, as long as there is an improvement, and therefore to explore each new neighborhood.

The efficiency (and efficacy) of ILS depends strongly upon the definition of the LocallyGenerate() function. If the neighborhood is too small[6], then the algorithm stalls, if it is too large, ILS behaves similarly to a PRS, losing all the advantages obtained introducing the LocallyGenerate procedure.

If the ILS is applied to $\mathcal{L}_f$, we obtain the algorithm known as Monotonic Basin Hopping (MBH), that is illustrated in Algorithm 4 and that resulted very effective in solving many differently optimization problems ([170],[86]) and it is the basic element of the works I present in Sections 2.2- 2.3 and Chapter 3.

Just as an illustration (a formal proof would ask more precise definitions and hypothesis), let us consider again the funnel function in Figure 2.5 and the result after applying an ideal local optimization in Figure 2.6. If the LocallyGenerate() function is defined

[6]smaller than the basin of attraction of the minimum inside the neighborhood

Algorithm 4 `Monotonic Basin Hopping`

```
1: procedure MONOTONIC BASIN HOPPING(N)
2:     x* =GloballyGenerate(), n = 0
3:     while n < N do
4:         z=LocallyGenerate(x*)
5:         z = L(z)
6:         if f(z) < f(x*) then
7:             x* = z, n = 0
8:         else
9:             n = n + 1
       return x*, f(x*)
```

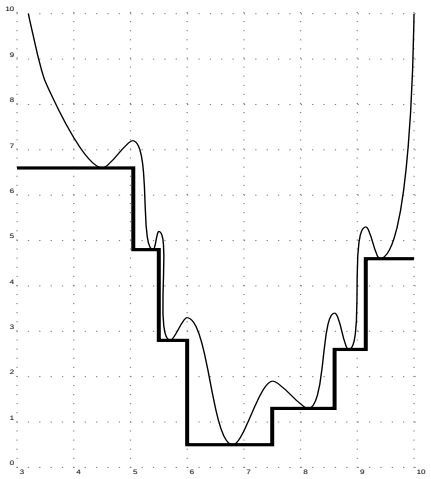


Figure 2.6: The effect of local optimization

correctly, i.e. if it defines a neighborhood such that it can contain the largest basin of attraction, and the stopping criterion allows to sample enough points in such neighborhood, then, MBH will converge to the global optimum with probability one. In fact, from any starting point in the domain there exists a descending (stair-case) path to the global optimum.

Some variants of MBH, that are quite general to be applied to different GO problems, but more flexible, are based on adapting the LocallyGenerate function to the current minimum point, similarly to Variable Neighborhood search for discrete optimization, as in Adaptive MBH methods ([109]). In [J26] and [J25], we proposed some methods that taking into account that MBH can be seen operating on the $\mathcal{L}_f$ function, try to determine search directions directly on the resulting step-function using a smoothing operator (see Figure 2.7). These methods proved to outperform the standard version of MBH and the Adaptative one on several difficult global optimization test functions (but they can result slower in terms of computational time).

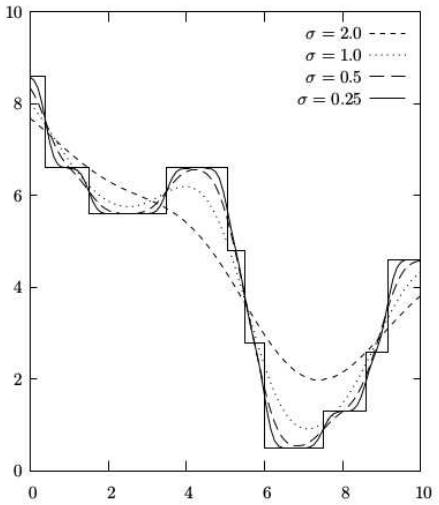


Figure 2.7: Example of different smoothing operators on $\mathcal{L}_f$

If the function does not present a single funnel, but multiples, as shown for example in Figure 2.8, the MBH strategy is not any more sufficient to have the guarantee to find the global optimum (even in probability). In fact, MBH can explore a single funnel and therefore can find only one of the different funnel bottoms (each one depending on the starting point).

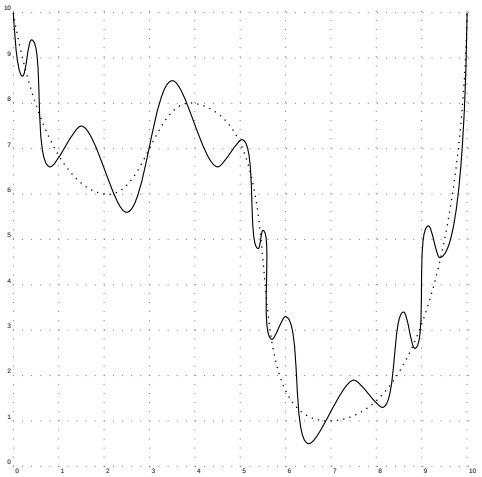


Figure 2.8: An example of a 2-"funnels" function

An *exploration* phase must be added to the method, that by construction is monotonically descending. Ideally, such phase must be able to allow exploring "completely" the search space. We mention some of the most common strategies that contain an exploration step:

- Repeating MBH from different randomly generated starting points. This strategy can be seen as the combination of a MBH and a Multistart where each "local search phase" of Multistart is performed as a MBH iteration.
- Simulated Annealing, eventually applied on $\mathcal{L}_f$. See Algorithm 5, where $P(\beta, z, x) = \min\{1, e^{-\beta(f(z)-f(x))}\}$ is a possible acceptance function, β^{-1} is commonly called temperature.

Algorithm 5 `Simulated Annealing`

```
1: procedure SIMULATED ANNEALING(β)
2:     x* =x=GloballyGenerate()
3:     while StoppingRule is false do
4:         z=LocallyGenerate(x)
5:         Extract a random probability P
6:         if P ≤ P(β, z, x) then
7:             if f(z) < f(x*) then
8:                 x* = z
9:             x = z
10:            Update β
       return x*, f(x*)
```

We can observe that if the temperature tends to zero we get exactly a PRS (or Multistart if applied on $\mathcal{L}_f$) and when it goes to ∞ we get an ILS method (or MBH).

- Using a Population-based MBH (PMBH): instead of working on one local solution at the time like in MBH, a "population" of K local solutions P (the so called "parents") is considered. Each of these solutions is perturbed and locally optimized to get a population of "children" C. This step can be seen as a single iteration of MBH using as current center each parent(see lines 4-5 of Algorithm 4). Then children are compared with parents to decide if they must be integrated in the current population. Differently from running K MBH in parallel, children are not necessarily compared with their own "direct" parent (the solution that was perturbed to get to them), but with the element in P that is closest to them. The idea behind this strategy is to try to compare solutions that are "close" and therefore allow the algorithm to locally explore each single funnel and at the same time keep a population that can represent well the overall search space (in some sense this strategy allow to counter-balance the intensification effect introduced by MBH for

the overall algorithm, and at the same time, for each sequence parent-children-parent enforces it). Therefore, it is necessary to define a "distance" function D between two solutions. For each child j, we search for the "least dissimilar parent"

$$n(j) = \arg \min_{i \in 1,\dots,K} \{D(C_j, P_i)\}$$

Let call $D(j) = D(C_j, P_{n(j)})$ the value of the minimum dissimilarity found and $\bar{D}$ the average distance in the current population P. At each iteration we decide if the new elements (children) must be inserted in the current population by one of the following criterias:

- if $D(j) < \bar{D}$, and C_j is better than $P_{n(j)}$: children C_j substitutes its closest parent $P_{n(j)}$;
- if $D(j) \geq \bar{D}$, then child C_j is sufficiently different: children C_j is added to the population P for the next iteration (or it substitutes the worst member)

We can imagine that, if function D is defined to recognize solutions belonging to the same funnel as similar, this strategy allows each element of the population to explore a single funnel. In particular, if an new element is found that is "enough" different from the current onces, it can be imagined that a new unexplored space region is detected.

In Sections 2.2-2.3, I will describe two family of problems (circle packing and space trajectory design) and how the presented ideas where adapted (and enriched) to solve some variants of them. The resulting methods' quality was proved by their capability to find new putative optima and to allow obtaining very good results in open optimization competitions.

2.2 Two circle packing problems

The problem of placing non overlapping objects belonging to $\mathbb{R}^d$ within a "smallest" container is a classical mathematical problem with important applications in manufacturing and logistics and, in particular, to problems related to cutting and packing – see e.g. [43] for a survey of applications in the fields of cutting, container loading, communication networks, facility layout.

The most studied case is the packing of *equal* circles in the unit square. Optimality proofs do not exist for the majority of cases, except of very small size problems. Therefore, to compare algorithm results it is necessary to have a common ground. For packing problem a reference is the E. Specht's web site [153] where putative optima for many different packing problems are reported, and in particular, for the packing of n equal circles in the unit square all best known solutions for up to 300 circles (at the time of our research, now it is up to 900) are reported.

To solve this problem many different approaches have been proposed as computer-aided optimality proofs [59, 137, 120], branch-and-bound approaches [112], and - due to the difficulty of solving exactly the problem - a significant number of heuristic approaches including a Multistart based on the unconstrained minimization of a properly defined energy function [136], a billiard simulation method [84], a two-phase approach with a first approximation phase based on local moves of circles along appropriately chosen directions and with an exponentially decreasing step-size and a second refinement phase based again on billiard simulation [28], an approach based on random perturbations of circles and possible acceptance of non improving moves [40].

Different variants of circle (and other 2-dimensional shapes) packing exist in the literature. Just to cite some of them in connection with our contribution, it is worth to mention the case where circles might have different sizes. In [156] the problem of placing circles with different sizes into a rectangular container with fixed width and minimum height is attacked by a combination of a branch-and-bound approach and a reduced gradient algorithm. In [92] a heuristic procedure is proposed to place circles with different sizes into a circular container with minimum radius, combining a quasi-physical approach (based on squeezing and collisions between circles) and a quasi-human approach (a circle compressed by many others, possibly including the external container, is moved to a new random position in order to get some "relief"). In [177] a build-up algorithm is proposed to sequentially place circles into a circular container. In [93] the minimization of the radius of the circular container is carried out through a combination of tabu search and simulated annealing ideas. Finally, in [43, 142] a global optimization approach has been tested.

In the following, two heuristic algorithm both based on MBH and PMBH methods will be described to solve two circle packing problems. The first has been defined for solving the packing of n equal circles in the unit square ([J21]), and the second for solving a "discrete" variant of circle packing proposed in an international competition based on unequal circles in a circular container ([J20]).

2.2.1 Packing n equal circles in the unit square

This packing problem is defined as follows: given a natural number n, find the maximum radius r that allows to place n non-overlapping circles of radius r inside the unit square. Different formulations are possible for such a problem, if we represent by x_i, y_i the coordinates of the center of the i-th circle, we can write:

$$\begin{aligned} r_n = \max\ & r \\ \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq r \quad & \forall\ i \neq j \\ r \leq x_i \leq 1 - r \quad & i = 1, \ldots, n \\ r \leq y_i \leq 1 - r \quad & i = 1, \ldots, n \end{aligned} \tag{2.3}$$

where r_n denotes the objective value of the problem for n circles.

Using variable scaling the solutions of the previous problem are equivalent to the solutions of the following formulation ([61]):

$$\begin{aligned} r_n = \max\ & r \\ (x_i - x_j)^2 + (y_i - y_j)^2 \geq r^2 \quad & \forall\ i \neq j \\ x_i, y_i \in [0, 1] \quad & i = 1, \ldots, n \end{aligned} \tag{2.4}$$

Formulation 2.4 is one of the most common forms used in numerical optimization methods, and the one we used in our experiments. The objective is linear, but constraints are reverse convex[7]. Thus this problem is nonconvex and highly multimodal. However, with respect to general problems with nonconvex constraints, for which even finding feasible solutions may be an extremely hard task, here feasible solutions are easy to find[8].

Monotonic Basin Hopping

In order to define a MBH strategy, a neighborhood structure (or equivalently a perturbation) has to be defined. Let $\xi_n = \frac{\alpha}{\sqrt{n}}$ (in our computation we set $\alpha = 0.5$). For each variable z, where $z = x_i$ or y_i, $i = 1, \ldots, n$, if we denote by $\hat{z}$ the current value of this variable, then the new value of the variable is uniformly sampled over the interval

$$[\max\{0, \hat{z} - \xi_n\}, \min\{1, \hat{z} + \xi_n\}] \subseteq [0, 1].$$

Some comments are needed at this point. First of all we need to comment our choice of the value for ξ_n. It is well known (see, e.g., [57]) that

$$r_n \sim \frac{2^{1/2} 3^{-1/4}}{\sqrt{n}}. \tag{2.5}$$

Then, let us consider an "active" pair of points of an optimal solution (a pair of points is said to be active if the distance between the two points is equal to r_n, i.e., if the

[7] the resulting feasible set represented by any of these constraints is the complement of a convex set, in this specific case the "outside" of a circle

[8] Sampling n points in the interval $[0, 1]^2$ and setting $r = 0$ produces a feasible solution

associated constraint in (2.4) is active). The maximum possible step-size allowed by our perturbation method is given by $\sqrt{2}\xi_n$ which is only slightly greater than $r_n/2$. This basically means that after the perturbation, with a very high probability, both points will still lie on the same side of a maximally separating line (i.e., a perpendicular line through the midpoint of the line segment connecting the two centers). This way the structure of the previous solution is still partially preserved, as it has to be for local moves. Also note that the perturbation of a point only affects its neighbors (it is easily seen that all points at distance larger than $r_n + 2\sqrt{2}\xi_n$ from a given point will still be at a distance larger than r_n after the perturbation, and therefore they will not have an impact on the objective function).

Next we note that there is some asymmetry in the perturbation of a point when this is close to the border. Indeed, in this case the perturbed point will be generated with a higher probability "towards" the interior of the unit square rather than "towards" the border (just think about the perturbation of a point lying at a vertex of the unit square). Therefore, this perturbation mechanism somehow acts as an implicit compression tool, which might have a positive effect. Indeed, for molecular conformation problems it has been observed that the insertion of explicit compression tools has extremely positive effects in detecting the most challenging global minima (see for example [113]).

Finally, we remark that our perturbation is still incomplete. Indeed, Problem (2.4) has a further variable which is r. This variable is perturbed in a different way with respect to the others. In order to generate a perturbed point within the feasible region of problem (2.4) the value for r has to be chosen in $[0, \bar{r}]$, where $\bar{r}$ is the minimum distance between points of the perturbed solution, i.e.,

$$\bar{r} = \min_{i \neq j} \sqrt{(\bar{x}_i - \bar{x}_j)^2 + (\bar{y}_i - \bar{y}_j)^2}. \tag{2.6}$$

It has been experimentally observed that setting $r = \bar{r}$ is not an efficient choice, while the choice $r = 0$ turns out to be quite efficient. This can be explained as follows. With $d = \bar{d}$ the perturbed solution lies on the border of the feasible region, while with $d = 0$ the perturbed solution almost surely lies in the interior of the feasible region, which gives more freedom to the local search procedure started from the perturbed solution.

Population MBH

From the computational experiments, it seems that larger instances present an higher number of different configurations. Following a successful approach developed for the optimization of molecular structures [87], we implemented a Population based version of our MBH. As already mentioned, in order to implement this strategy, a measure of dissimilarity is needed. Inspired by the ideas successfully used for molecular clusters in [105], we defined a dissimilarity function based on the number of "neighbors" of each circle in a given solution. Let δ be a threshold and represent with $X = \{X_i \in \mathbb{R}^2\}_{i=1}^n$ the coordinates of the centers of the circles in a given disk packing. Given a disk i, let define the number of disks whose distance from disk i does not exceed δ:

$$N_\delta(X, i) := |\{j \in 1, \ldots, n, j \neq i : \|X_i - X_j\| \leq \delta\}|$$

We can count how many circles has k neibourghs, using histograms:

$$H_\delta(X,k) := |\{i \in 1,\ldots,n : N_\delta(X,i) = k\}|$$

Then we can compare two packings X and Y evaluating

$$D_\delta(X,Y) = \sum_{k=0}^{n}(k+1)|H_\delta(X,k) - H_\delta(Y,k)|.$$

Thanks to the weighting factor $k+1$, more weight is given to configurations in which the number of disks with many neighbors is different. As a general rule it is advisable to choose a value for δ which is somewhat larger than the expected minimum distance between pairs of points. In fact, in many configurations, even putative optimal ones, there are "free disks" (a.k.a. "rattlers"), i.e., disks that can be moved without violating any constraint. Therefore, these disks might or might not be in contact with others, in different equivalent solutions, so that through a dissimilarity measure (with a too small δ) two configurations which differ only for the location of the rattlers might erroneously be considered as different. In our computations we usually set $\delta = 1.5/\sqrt{n}$.

This measure of dissimilarity is quite easy to compute and displays a very good discriminating power; moreover it is invariant to rigid transformations of the packing configurations, for example, given any solution and the equivalent one that is obtained swapping coordinates x with coordinates y ($\pi/2$ rotation) present a dissimilarity of zero.

A short summary of the computational results

In our computational experiments, for the standard version of MBH we employed $N = 100$ (maximum number of consecutive iterations without a success), while for the population variant (PBH) we choose 40 elements in the population and $N = 30$; in PBH we used this parameter on the whole population, i.e., we choose to stop the algorithm as soon as 30 generations of children had been obtained with no improvement in the best one. For local optimization, we used SNOPT[9] [82] with feasibility tolerance parameters set to 10^{-12}.

As already mentioned, putative optimal configurations are maintained by E. Specht on his web site [153]; there, after a new finding for $n = 97$ in January 2003, Specht comments: *"It seems to be very unlikely to find still better packings for $n \leq 100$, but it is possible"*. Basically, since many different methods had been tested on these instances, it seemed to be reasonable to conjecture that packings known at that time for n up to 100 were indeed the optimal ones. Therefore, in order to check the robustness of our method the first aim was to show that we were able to efficiently reproduce at least all the best known packings up to $n = 100$. What we obtained was actually much more than that. Indeed, in this range we have been able to detect many new best known solutions, now reported in [153] and to confirm most of the remaining. Stimulated by this success we performed numerical experiments up to $n = 130$ again obtaining many improved

[9] a large scale SQP method

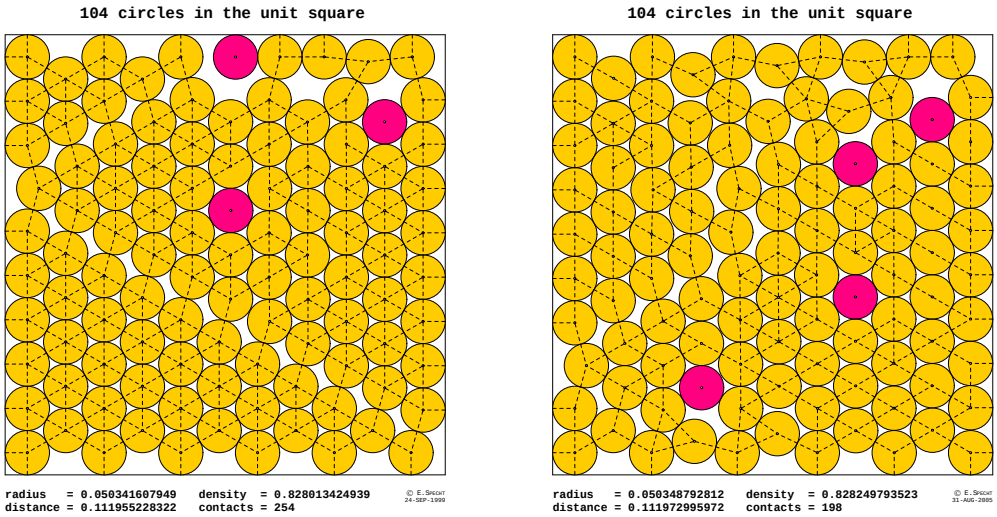


Figure 2.9: Putative optimal packings for $n = 104$: previously known record (left, $d = 0.111955228322$) and new putative optimum (right, $d = 0.111972995972$)

putative optima. In the Table 2.1, the results for $n \in [50, 130]$ at the time of writing our paper [J21] are reported: in normal typeface the values of n for which we could obtain the previously known putative optimum [10]; in **bold** the values of n corresponding to cases in which we could improve over previously known putative optima and, finally, in *italic* those instances for which we were not able to obtain, within the prescribed accuracy, the putative optimum. We remark that in this problem an apparently small improvement in the distance can, and usually does, imply a significantly different geometry. Just for illustration we present in Figure 2.9 a comparison between the previous putative optimum and our improved configuration for $n = 104$. In this figure, dark disks are "rattlers".

Many of the new records and (when no new record has been detected) of the old putative optima have been found several times by both MBH and PMBH. For what concerns failures, only two cases have occurred. $n = 112$ could not be found even after running 100 PBH independent experiments. The case $n = 75$ is particularly significant: we could indeed find a configuration whose distance is just 10^{-10} smaller than that of the putative optimum, but most of the times we reached a configuration whose distance is more than 10^{-7} smaller than that of the putative optimum. In order to check whether the optimal configuration could have been recovered by means of local perturbations, we adopted the technique described in [64] to obtain a rigid movement of disks which eventually led to an "unjamming" of our best configuration. From the resulting configuration of this procedure a single run of a local optimization method led us to the putative optimal solution for $N = 75$. The same procedure was however not successful for $N = 112$,

[10] with a maximum error on the distance of 10^{-12}

50	51	52	**53**	54	55	56	57	58	**59**
60	61	62	63	64	65	66	67	**68**	69
70	71	72	**73**	74	*75*	76	**77**	**78**	79
80	81	82	83	84	**85**	**86**	87	**88**	89
90	91	92	93	94	**95**	96	97	**98**	99
100	**101**	102	103	**104**	**105**	**106**	**107**	**108**	**109**
110	**111**	*112*	**113**	114	**115**	**116**	**117**	**118**	**119**
120	**121**	**122**	**123**	124	125	126	**127**	**128**	**129**
130									

Table 2.1: Results obtained by executing 100 independent experiments with MBH for $n \leq 100$ and 10 independent runs with PMBH for all values of n.

which thus remains the unique failure of our method for $N \leq 130$.

Some observations on the formulation and random generation

We observe that solving problem (2.4) is equivalent to solving any problem with the following form:

$$\begin{aligned} &\max \ f(r) \\ (x_i - x_j)^2 + (y_i - y_j)^2 \geq g(r^2) \qquad &\forall\ i \neq j \\ x_i, y_i \in [0,1] \qquad &i = 1, \ldots, n \end{aligned} \tag{2.7}$$

where f is a monotonic increasing function and g is some continuous function satisfying

$$g(r^2) \begin{cases} = r^2 & \text{for } r \geq r^\star \\ \leq r^2 & \text{for } r < r^\star \end{cases}$$

where $r^\star$ denotes a lower bound for the optimal value of (2.4). Although any function f and g satisfying the above conditions leads to formulations which are equivalent from the theoretical point of view, considerable differences can be observed in the practical behavior.

Furthermore, we tested the influence on performance of the addition of a simple explicit compression mechanism: after having perturbed a solution, we further modify this solution by multiplying all of the coordinates by a factor $c \in (0,1)$ (in particular, we tested $c = 0.8$); thus the resulting configuration shrinks to fit within the square $[0,c]^2$.

In order to test the impact of different choices, we selected four test environments (listed in Table 2.2) and performed for each of them 100 runs with n ranging from $n = 31$ (the first n value for which a computer-aided proof of optimality has not been given yet) up to $n = 100$. Note that choice C1 is equivalent to (2.4) and that the introduction of the *tol* value is necessary to guarantee the differentiability of the objective function at $r = 0$. Also note that at each iteration of a MBH run, the $r^\star$ value is the current record value for that run.

Choice	f	g	Compression
C1	r	r^2	No
C2	$100r$	$r^2 - [\max\{0, r^\star - r\}]^2$	No
C3	$100\sqrt{r + tol}$	$r^2 - [\max\{0, r^\star - r\}]^2$	No
C4	$100\sqrt{r + tol}$	$r^2 - [\max\{0, r^\star - r\}]^2$	Yes

Table 2.2: Different tested choices ($tol = 10^{-5}$).

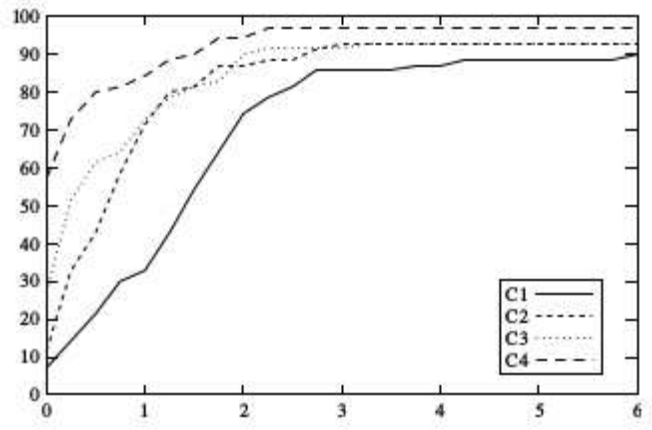


Figure 2.10: Performance profiles of the four tested choices. Test choice C4 is the best one in 57% of the problems while in 84% cases its success rate is at least 1/2 with respect to the best one.

In Figure 2.10, we report – in a very compact form – results for the different variants in the form of performance profiles, introduced in [63] with the aim of comparing different solution algorithms (let us represent it by set $\mathcal{C}$) on a given test set ($\mathcal{P}$). In what follows, we give a brief explanation of how to interpret them. Let us define a performance measure perf(c, p) for each algorithm (strategy) c and for each instance (problem) p. Examples can be: the total running time, the value of the best solution found. In our case, we consider the putative optimal radius $r_p(c)$ found by choice c on instance p and define the following "counting" function:

$$\text{Perf}_c(x) = \frac{100\|p \in \mathcal{P} : r_p(c) \leq 2^x \cdot \min_{i \in \mathcal{C}} \quad r_p(i)\|}{\|\mathcal{P}\|} \tag{2.8}$$

This curve represents the percentage of problems where choice c has a performance that is not worst than 2^x times with respect to the best choice. Therefore, on the ordinate it can be read the percentage of instances where the algorithm finds the putative optima, for $x = 1$ the number of tests were the gap with the putative optima is smaller than or equal to 100%, and so on. Very roughly speaking, up-left positioning in the figure is a sign of good performances.

We can observe that C4 strategy is the most effective, showing that an "intelligent" choice of the starting points (in this case the utilization of a compressor factor) is a key element, doubling the number of successes with respect to the same formulation without compression (C3). Also the choice of a different formulation (from C1 to C2, and from C2 to C3) has an impact, increasing not only the number of successes (twice from C2 to C3), but also the quality of the overall local solutions (not only the putative one).

2.2.2 Circle packing contest

In this section we present the solution strategy we used to tackle the circle packing competition proposed in the "Al Zimmermann's Programming Contests"[180]: given a positive integer n, place n disks of radii, respectively, equal to $1, 2, \ldots, n$ inside a circle whose radius (r_n) is minimal. The participants were asked to propose solutions for all n in $5, \ldots, 50$, and a total score was assigned to each participant considering their results for each given size. More precisely, for each submitted solution of size n, a quality measure is defined:

$$Q_n = \frac{\text{best} \quad r_n}{\text{submitted} \quad r_n} \tag{2.9}$$

where best r_n represents the value of the best found solution for size n among all participants and no submission for a given n produced a value $Q_n = 0$. The total score was given by:

$$\sum_{n=5,\ldots,50} (0.7Q_n + 0.1(Q_n^{16} + Q_n^{128} + Q_n^{1024})) \tag{2.10}$$

Therefore, the maximum attainable score was of 46 ($Q_n = 1 \forall n = 5 \ldots 50$).

Let consider a mathematical programming formulation for the problem:

$$\min R \tag{2.11}$$

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq i + j \qquad \forall\, 1 \leq i < j \leq n \tag{2.12}$$

$$\sqrt{x_i^2 + y_i^2} \leq R - i \qquad \forall\, 1 \leq i \leq n. \tag{2.13}$$

where, as in the previous problem, x_i, y_i denote the coordinates of the center of circle i. Constraints (2.12) force circles i and j not to overlap, while constraints (2.13) force each circle to be included in the disk with radius R centered at the origin. Although the objective is linear and constraints (2.13) define a convex region, constraints (2.12) are reverse-convex (as for the previous packing problem).

In comparison to the problem of placing identical circles in a smallest container, here the fact that each disk has a different radius generates new difficulties and challenges. In fact, while the problem with identical radii can be considered as a pure continuous optimization problem, here the fact that each circle has a different radius adds some sort of combinatorial structure over the original one. Moreover, although a direct application of this model to real life problems is difficult to imagine, nonetheless the ideas used in attacking this problem might find an application in methods for optimal placement of figures with the same shape but different sizes in a container.

Strategies

We developed several models and algorithms during the course of the contest. Many of them were discarded as non productive, just a few of them survived until the last weeks and gave us the possibility of winning against 154 competitor teams. In the following, we outline the main ideas underlying most of the models and methods we tried.

MBH Given our experience in the field of molecular conformation problems [65] and in equal circle packing [J21], we decided to try a similar approach also in this case.

An initial solution was found by drawing the coordinates of the center of each circle from a uniform distribution on a suitably large box[11]. After the generation of circle centers, it proved useful, sometimes, to re-scale them in order to avoid circle overlapping.

We tested different perturbation rules but ended up with the following: two circles, with similar radii (the difference between the radii could not be larger than 2), were chosen at random and their radii exchanged. This is a combinatorial neighborhood exploration and it proved to be much more efficient than, e.g., continuous random displacement of the circle centers. After swapping two circles either we restored feasibility through re-scaling or not (usually 50% of the times).

A critical issue both for the initial generation and for the perturbation was related to the choice of the initial value for the R variable. Our local optimizer, SNOPT [82], was extremely sensitive to the initialization of the R variable prior to local optimization (the same has been observed for other nonlinear solvers). Choosing to initialize R in the most natural way, i.e. the radius of the smallest circular container containing all circles, gave the initial solution too much rigidity and, often, the local optimizer did not produce any useful solution (similarly to what we obtained for the packing of equal circles in a square). We tried different other possible values for R and the best choice turned out to be setting R slightly (say 10%) larger than the radius of the smallest circular container containing all circles. This strategy can be considered similar to the compression technique we presented for the packing of equal circles in the square.

Reduction of the space of variables It was quite early observed that in putative optimal configurations of N circles, the smallest ones did not play any role and very often they were "rattlers", i.e. disks which could be displaced to a different position without violating any constraint nor worsening the objective. Thus we decided to solve a reduced optimization problem in which instead of placing disks of radii $1, \ldots, N$, we only looked for optimal configurations of $N-m$ circles of radii $m+1, \ldots, N$, thus discarding the first m. The value of m we chose was usually around $0.25N$. Removing smallest circles had three beneficial effects:

- first, the dimension of the feasible space is reduced, which make local optimization slightly easier and quite significantly faster;

[11] usually chosen as $[-N^2/2, N^2/2]$

- second it allows a greater freedom in choosing the positions of the remaining disks, which otherwise could be artificially constrained by the position of small circles which could have found a place somewhere else without any difficulty;
- finally, we observed that removing smaller disks has the effect of reducing the number of local optima – in particular, all locally optima packings which differ only in the position of the smallest circles are reduced to a single one.

Obviously, after optimization with MBH, we had to restore the eliminated circles, if possible, without enlarging the container radius. This was done by sequentially re-inserting missing disks, one at a time starting from the largest one (see [J20]).

Backward and forward moves Since optimal configurations for close N values might be quite similar, quite regularly we also tried backward and forward moves in search of improvements. In particular, while looking for an optimal configuration for N circles, we tried to start from a known solution with $N+1$, in which we eliminated the smallest circle (or the $m+1$ smallest ones in order to have only $N-m$ circles as described before) and reduced all the radii by one. Analogously, for forward moves, we started from a configuration of $N-1$ circles, removed the $m-1$ smallest circles in order to have only $N-m$ circles, augmented all the radii by one, re-scaled in order to have a feasible starting solution. The resulting configurations were inserted as the "good enough" ones within the population-based approach (see next point).

Population based approach Another strategy we implemented with success was that of working with a population of solutions. As before, let us represent with $X = \{X_i \in \mathbb{R}^2\}_{i=5}^{n}$ the coordinates of the centers of the circles in a solution, therefore, X_i represents the coordinates of the circle of radius i. Letting ρ_i^X be the distance from the origin to the center of circle i in the solution X. Then given two solutions X and Y, we defined the following pairwise dissimilarity measure:

$$D(X,Y) = \sum_{k=\lceil N/2 \rceil}^{N} k|\rho_k^X - \rho_k^Y|$$

This measure takes into account only the largest circles and is based on their distance from the center of the container. Furthermore, it gives more weight to different positions of larger circles, as this is a good indicator of significantly different structures. Note that this dissimilarity measure is invariant under rotation.

The initial population was mostly generated at random; however we chose to generate one of its elements as an already "good" one. In particular, this element might be: (a) *the current record*; (b) *the result of a MBH run*; (c) *the result of a backward or forward move* (as described previously).

Inserting a "good enough" element in the initial population turned out to be an extremely beneficial choice. Indeed, we observed that the population is very quickly filled in with neighbours of this "good enough" element and the population-based

strategy allows to explore more evenly the state space of possibly worse but still good configurations around this element, avoiding the danger of being too greedy. We observed, in several occasions, that improvements were consequences of the phenomena of backtracking (a child is worse than its father but enters the next population) and survival (a father is worse than its child but survives in the next population) as observed for molecular conformation problems (see [87] for more details).

Note that, while using as "good enough" element the current record or the result of forward and backward moves does not guarantee a full exploration of the search space, using the results of MBH with random restarts guarantees a complete exploration of this space.

In Table 2.3 we report all the best results obtained during the competition together with our results and, within parentheses, the ranking of our result with respect to all the others for each size. The list of all records as well as their coordinates can be obtained from the contest web site [180].

n	Best	Our	Rank	n	Best	Our	Rank
5	9.00139774	9.00139774	**1**	29	99.51231790	99.51231790	**1**
6	11.05704039	11.05704039	**1**	30	104.57855508	104.74779928	3
7	13.46211067	13.46211067	**1**	31	109.77194698	109.77194698	**1**
8	16.22174667	16.22174667	**1**	32	114.86543833	114.90022384	2
9	19.23319390	19.23319390	**1**	33	120.21695714	120.24932984	2
10	22.00019301	22.00019301	**1**	34	125.43350175	125.61242286	2
11	24.96063428	24.96063428	**1**	35	131.15635463	131.16650268	2
12	28.37138943	28.37138943	**1**	36	136.53490083	136.53490083	**1**
13	31.54586701	31.54586701	**1**	37	142.17498053	142.25633406	2
14	35.09564714	35.09564714	**1**	37	142.17498053	142.25633406	2
15	38.83799550	38.83799550	**1**	38	147.85769135	147.99157574	2
16	42.45811643	42.45811643	**1**	39	153.55530119	153.60382813	2
17	46.29134211	46.29134211	**1**	40	159.48902487	159.57390300	3
18	50.11976262	50.11976262	**1**	41	165.29190968	165.29190968	**1**
19	54.24029359	54.24029359	**1**	42	170.92576161	170.92576161	**1**
20	58.40056747	58.40056747	**1**	43	177.07434007	177.23962454	2
21	62.55887709	62.56005858	2	44	183.17606157	183.37007125	2
22	66.76028624	66.76028624	**1**	45	189.63543910	189.67917387	2
23	71.19946160	71.19946160	**1**	46	195.91076339	195.91076339	**1**
24	75.75270412	75.75270412	**1**	47	202.18561174	202.22801940	2
25	80.28586443	80.28586443	**1**	48	208.63594672	208.63594672	**1**
26	85.07640122	85.10628281	3	49	214.66195201	214.66195201	**1**
27	89.79218156	89.82957381	4	50	221.08975259	221.08975259	**1**
28	94.54998647	94.70586218	3				

Table 2.3: Best and our solution values, plus our ranking with respect to all submitted solutions

2.3 Space mission analysis

A very important problem in mission analysis is the planning of interplanetary trajectories, which consists in launching a spacecraft from a given astronomical body (usually the Earth) along a trajectory which leads to some other astronomical body. The aim of the mission can be to land on the body or to put the spacecraft into the planet's orbit and the objective of a model is to help trajectory planners in taking the best decision, e.g., on the starting date and other relevant parameters, in order to obtain a "low-cost" mission.

In Figure 2.11 is illustrated the real trajectory of the mission Rosetta, a probe build by ESA and launched on the 2th of March 2004. The probe collected data during several fly-bys [12] and finished its mission by hard-landing on the comet 67P/Churyumov-Gerasimenko the 10th November 2014. The full lines (in color) represents the orbits of the different astronomical bodies (planets/asteroids/comet) encountered during the trajectory, the dotted lines represent the trajectory itself.

In general, accurate models are computationally very demanding. For this reason simplified models are typically used in the first planning phase: it is assumed that the spacecraft is equipped with a chemical propulsion engine and the aim is that of minimizing the overall energy spent during the mission.

Global optimization techniques are usually applied to these simplified models. These might return a set of good solutions, that can be further refined through more accurate and costly models.

Different papers (see, e.g., [2, 78, 101, 148]) propose Genetic Algorithms (GA) for this kind of problems. Evolutionary strategies, in particular Differential Evolution (DE), turn out to be a valid alternative (see, e.g., [58, 139]). Hybrid methods have also been proposed. In these methods either the problem structure knowledge (see [98]), or the results of some optimization methods (see [167, 165, 166, 163]) are exploited to evaluate portions of the feasible region and, consequently, to discard them or, alternatively, to intensify the search within them. Particularly relevant are recent studies carried on to compare the performance of different GO algorithms on different benchmark problems. The tested algorithms in these studies (see [60, 96, 97, 132, 163, 164, 168]) include, besides GA and DE, also Particle Swarm Optimization, Adaptive Simulated Annealing, GLOBAL, COOP, Multilevel Coordinate Search, DIRECT.

It is worth to note that, during last years, people at ESA ACT (Advanced Concept Team) carried on a considerable effort to make standard models of many benchmark problems (see `http://www.esa.int/gsp/ACT/inf/op/globopt.htm`), available both in C/C++ language and in MATLAB.

The currently available studies show that DE often performs quite well. However, a recent study [164] reveals that a basic version of Monotonic Basin Hopping (MBH, see, e.g., [104, 111]) is able to outperform some other algorithms, including DE, on some benchmark problems. This fact led us to propose a method based on the MBH approach.

[12] parts of the trajectory where the probe is close enough to an astronomical body

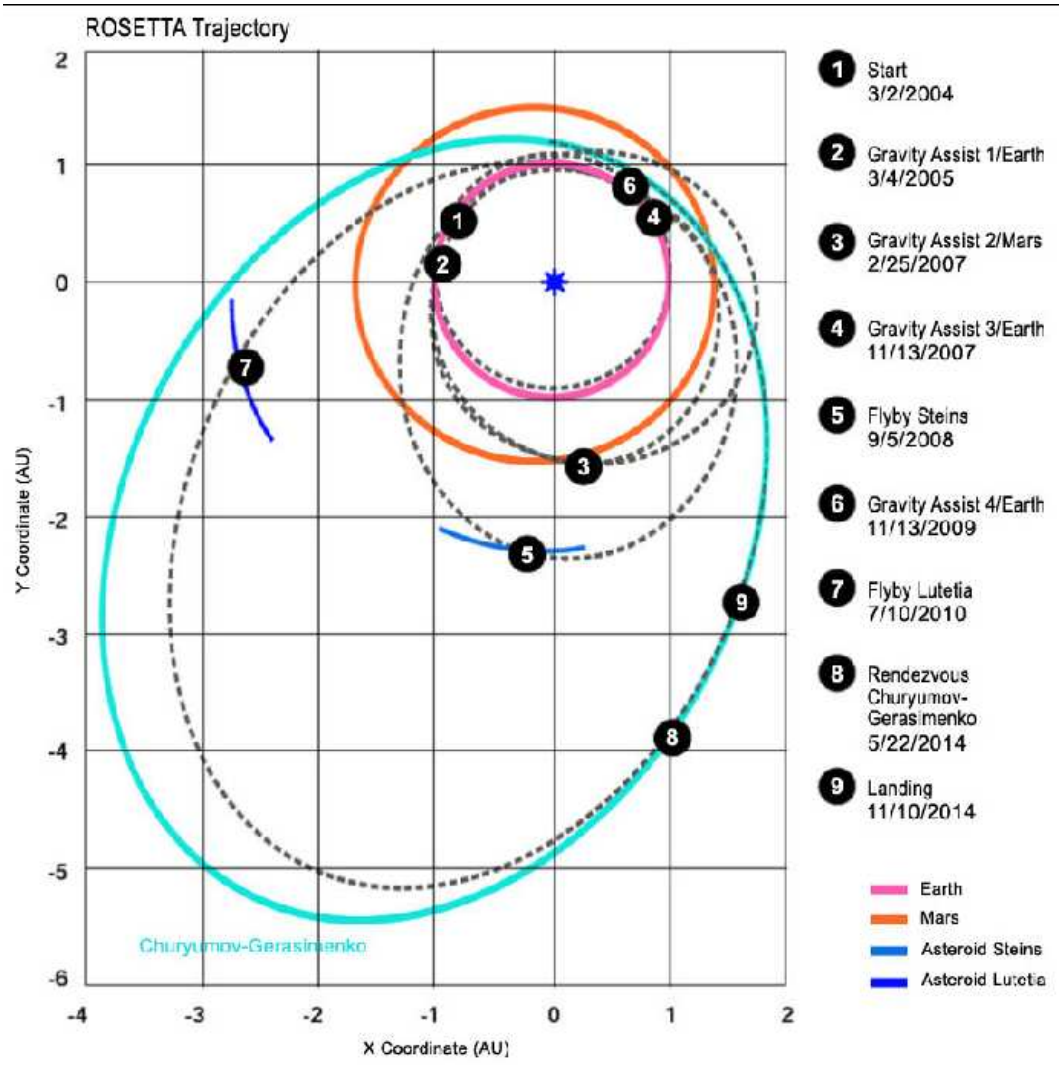


Figure 2.11: Trajectory of mission Rosetta

Before introducing the mathematical formulation and our method, we need to introduce some definitions.

Leg A *leg* is the trajectory followed by the spacecraft between two astronomical bodies (planets or asteroids). For example, the dotted line between number 2 and 3 represent a leg between the Earth and Mars.

Pericenter radius The *pericenter radius* at an astronomical body is the minimum distance between the trajectory of the spacecraft and the body (see Figure 2.12)

Swing-by A *swing-by* or *gravity assist* maneuver is the result of the gravitational interaction between the spacecraft and the astronomical body: as the spacecraft gets

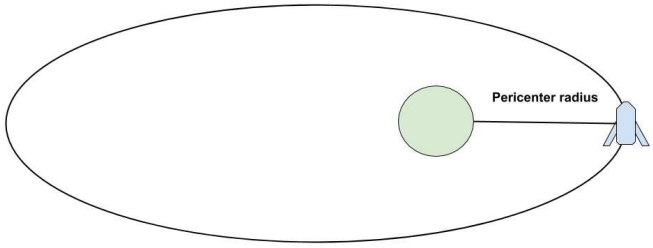


Figure 2.12: Pericenter radius

close to the body, such interaction does not change the modulus of the spacecraft velocity but changes its direction. The new direction depends both on the modulus of the velocity and on the pericenter radius. In our example, we have 4 swings-by: Earth (4/03/2005), Mars (25/02/2007), Earth (13/11/2007), Earth (11/13/2009).

Lambert's problem and arc Given two points in space and the time of flight between them, the trajectory followed by the spacecraft between the two points can be calculated by solving a *Lambert's problem*, which basically consists in the solution of a second order ordinary differential equation with boundary conditions. The resulting trajectory is called a *Lambert's arc*. Lambert's problems usually have multiple solutions but, exploiting some problem knowledge (and, actually, accepting the risk of excluding good solutions), we can introduce further assumptions which reduce the number of solutions to one (we refer, e.g., to [98] for more details).

Deep Space Maneuver A *Deep Space Maneuver* (DSM in what follows) is a change in the spacecraft velocity during a leg (the spacecraft is usually assumed to be able to thrust its engine at most once during each leg). When a DSM is added inside a leg, the leg can be considered as split in two Lambert's arc.

2.3.1 Problem definition and analysis

In the problems discussed in this paper we have a sequence of $n+1$ astronomical bodies $B_0, \ldots, B_n$ (B_0 is usually the Earth). The bodies are not necessarily distinct. We would like to visit the sequence in such a way that the overall energy consumption is minimized. Note that in what follows it is assumed that the sequence is *fixed*, but we may also think of models where the sequence of bodies is part of the decision problem, thus implying the introduction of discrete choices in the models.

MGA problem

The first model we consider is the Multiple Gravity Assist (MGA in what follows) model. In this model the variables are:

- t_0, the starting date of the mission;

- $T_i,\ i = 1, \ldots, n$, the time of flight along leg i (joining body B_{i-1} with body B_i).

Given the values of these variables, we are able to identify the positions p_{i-1} and p_i respectively of body B_{i-1} at time $t_0 + \sum_{j=1}^{i-1} T_j$, and of body B_i at time $t_0 + \sum_{j=1}^{i} T_j$. Therefore, the solution of the corresponding Lambert's problems allows us to identify the Lambert's arcs (which are conic arcs, either part of an ellipse or of an hyperbola) along all legs. It is then possible to compute the velocities at the end of each leg i and at the beginning of the following one $i + 1$, $i = 1, \ldots, n - 1$. In order to transfer from one leg to the next one, the spacecraft needs to provide a single impulse, a single change of velocity denoted by Δv_i. In the initial leg the spacecraft will have to provide a single impulse Δv_0 to leave the starting planet's orbit and reach the starting velocity at the initial leg. Similarly, in the final leg the spacecraft will have to provide a single impulse Δv_n to move from the final velocity in the last leg to the velocity of the target astronomical body B_n. Each impulse causes an energy consumption proportional to the modulus of the change of velocity. Therefore, in order to minimize the overall energy consumption, we are led to the following objective function:

$$\|\Delta v_0\| + \sum_{i=1}^{n-1} \|\Delta v_i\| + \|\Delta v_n\| . \tag{2.14}$$

Usually, MGA models also include constraints on the pericenter radius r_i at each intermediate body B_i, $i = 1, \ldots, n - 1$, which typically require r_i not fall below a given threshold r_i^{min}: the spacecraft needs to be far enough from body B_i in order to be able to leave the planet orbit and avoid a "forced" landing, due to the gravitational force. Such constraints can be kept as explicit ones or, alternatively, can be moved to the objective function, through the addition of properly defined penalization terms in (2.14). Note that the pericenter radius at each intermediate body is a dependent variable in the MGA model: once we have computed the Lambert's arcs, we can also derive it through appropriate formulae.

MGADSM problem

The second model allows for the introduction of DSMs and will be denoted by MGADSM. Such model is more flexible but also more complex. In particular, it requires the introduction of new variables besides those already discussed for the MGA model, in order to take into account the DSMs:

- the modulus and the direction (defined by two angles) of the spacecraft relative velocity at the initial body B_0 (V_∞, u, v)[13];
- the time instant in which each DSM maneuver takes place; usually these are formulated through the introduction of variables $\eta_i \in [0, 1]$, $i = 1, \ldots, n$ which define

[13]For the sake of precision, although related to angles, variables u and v are usually constrained to belong to the interval $[0, 1]$: they represent linear transformations of the polar coordinates of the spacecraft relative velocity at the initial body B_0 (see, e.g., [164, 168] for details).

at which portion of leg i the DSM maneuver occurs. More precisely, during leg i a DSM is performed at time

$$t^i_{DSM} = t_0 + \sum_{j=1}^{i-1} T_j + \eta_i T_i.$$

This way, instead of having a unique Lambert's arc during leg i, we have two of them, the first one joining the position (at time $t_0 + \sum_{j=1}^{i-1} T_j$) of body B_{i-1} with the position of the spacecraft at time t^i_{DSM}, and the second one joining the position of the spacecraft at time t^i_{DSM}, with the position of body B_i at time $t_0 + \sum_{j=1}^{i} T_j$;

- the pericenter radii r_i, $i = 1, \ldots, n-1$, at the intermediate bodies are now independent variables;
- finally, at each intermediate body B_i, $i = 1, \ldots, n-1$, we need to choose an angle b_i. The spacecraft's incoming velocity and the orbit's eccentricity of the Lambert arc joining the position of the spacecraft at time t^i_{DSM}, with the position of body B_i at time $t_0 + \sum_{j=1}^{i} T_j$, allow to define a cone, along whose surface the spacecraft's outgoing velocity lies: the value of angle b_i uniquely identifies the direction of such velocity along the cone's surface.

Each DSM requires a change of velocity and this implies an energy consumption which has to be included in the objective function.

All these models can be considered as **box-constrained black-box** ones and, like we did in this paper, no information on the problem, like, e.g., analytical derivatives, can be exploited.

2.3.2 A MBH for space trajectories planning

The core elements to define a MBH strategy are the perturbation (LocallyGenerate function) and the standard local search procedure (represented by $\mathcal{L}_f$). In theory, any standard local search method can be used to obtain $\mathcal{L}_f$, however this choice can change greatly the practical behavior of the method. We can further enhance the performance of the overall method by employing a "two-phase" local search. The first phase is aimed at driving the search towards promising portions of the feasible region (the result may not even be a local minimizer), while the second phase is a refinement one, actually leading to a local minimizer. In this sense, a two phase local optimization can be interpreted in two complementary ways: as a transformation of function $\mathcal{L}_f$, aiming at increasing the probability to reach the global optimum (introducing some global information) or, as a modification of the random generation/perturbation, i.e. the first local search can be seen as a refinement step of the random generation, moving from an "uniform" generation to one ideally concentrated near by the global optimum (or good local optima). For example, in molecular conformation problems the first phase is defined through the addition of geometric penalization terms to the objective function (see, e.g., [65],[O9]) driving the search through more compact solutions (that seem to be good candidates

for the global solution). Of course, the definition of a two-phase approach is strictly problem dependent and must be adapted consequently.

Local optimization inspired by Implicit Filtering Even if almost everywhere smooth, space trajectory problems usually display, even for small instances, a function landscape which is characterized by an enormous number of local optima: the objective function looks like a smooth one, perturbed by some sort of noise which generates many local optima. Examples can be seen for example in [132], in which bi-dimensional plots for the objective function in the case of Earth-Jupiter-Saturn, Earth-Mars and other missions, show the presence of many local optima often clustered in a periodic structure.

An interesting technique, called *implicit-filtering algorithm* (IF), has been introduced by Kelley [83, 135] for the case of box-constrained problems for which the objective function can be thought as $f(x) = f_s(x) + \phi(x)$, where $f_s(x)$ is smooth and $\phi(x)$ is not differentiable but such that $f_s(x) \gg \phi(x)$. The basic idea is to substitute the exact gradient with a finite difference estimate and use it inside a standard descent algorithm for smooth optimization. The approximated gradient is computed by forward or centered finite differences, with a step h which, starting from a relatively high value, is gradually decreased during the iterations; this way convergence properties can be derived (see [53]).

We decided not to actually implement an IF algorithm but to mimic in some sense its behavior. We chose to use the SNOPT [82], which makes available a finite difference support in which either forward or central differences are used adaptively, applying the following formulae:

Forward scheme

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(\hat{x}) - f(x)}{h(1 + |x_i|)} \quad \text{where } \hat{x}_j = \begin{cases} x_j & j \neq i \\ x_i + h(1 + |x_i|) & j = i \end{cases} \tag{2.15}$$

Central scheme

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(\hat{x}^+) - f(\hat{x}^-)}{h(1 + |x_i|)} \quad \text{where } \hat{x}_j^{\pm} = \begin{cases} x_j & j \neq i \\ x_i \pm \frac{1}{2}h(1 + |x_i|) & j = i \end{cases} \tag{2.16}$$

In SNOPT, forward differences are used by default, using a step length parameter h which can be chosen by the user; SNOPT switches from forward to central difference when the current point is close to a stationary point. Moreover, the algorithm reduces the step length to ensure feasibility with respect to the linear and box constraints.

The main advantages of this strategy are the following:

- forward differences require half function evaluations with respect to centered ones;
- step length reduction occurs only to ensure feasibility or close to the solution.

SNOPT with finite difference is not an IF-like algorithm, as the strategy used is that of performing very precise derivative estimation, trying to avoid excessive function evaluations. However, as previously suggested, we can separate the local search in two phases: during the first one we carry on an "imprecise" search by choosing a large step h; during the second phase we refine the result by switching to a much smaller h value. Such two-phase local search resembles the behavior of an IF algorithm.

On the random generation and variable scaling Some choices turned out to be quite beneficial both in terms of effectiveness and of robustness.

Variable scaling According to our experience, confirmed also by other works like, e.g., [20], it is important to scale variables in order to reduce numerical problems: each ESA ACT test problem is composed by variables with widely different scales: variables in fact include angles, velocities, departure dates, and so on. In our method we scaled each variable to the interval $[-0.5, 0.5]$.

Periodic variables Many test problems contain variables representing angles constrained to lie in $[0, 2\pi]$. Clearly we can discard such limitation w.l.o.g. and let variables be free, but, to comply with the previous issue, before any local optimization phase variables are scaled back to the original interval. This way we obtain the following advantages:

- the local optimizer can explore the solution space more freely when close to the former boundary;
- during the perturbation step we have not to deal with feasibility of the perturbed variables.

We performed a large set of numerical experiments with two objectives in mind: first we wanted to obtain good trajectories, i.e. solutions which were comparable or possibly better than those deposited at the ESA ACT web site. Second, we wanted to check which of many possible variations in our algorithm were the most successful and the most robust ones. By this last term we mean that one of our aims has been that of proposing an algorithm which was capable, in many cases, to produce a set of good solutions.

We briefly discuss here our results only for MGADSM problems characterized by the presence of box constraints only. The meaning of each variable is briefly summarized in Table 2.4, while problem characteristics (number of variables and sequence of astronomical bodies visited) are listed in Table 2.5. Note that Tandem is not a single problem but a set of 24 problems, each corresponding to a different sequence of planets from the Earth to Saturn.

Table 2.4: Variables for box constrained ESA MGADSM problems

Name	Meaning	#
t_0	departure time	1
V_∞	dep. vel. modulus	1
u	dep. vel. angle1	1
v	dep. vel. angle2	1
T_i	time of flight	n
η_i	time of DSM i	n
rp_i	pericenter radius at swing-by i	$n-1$
b_i	outc. vel. angle at swing-by i	$n-1$

Problem Name	variables	n	Planet sequence
Cassini	22	5	E V V E J S
Messenger	18	4	E E V V Me
Rosetta	22	5	E E V E E 67P
Tandem	18	4	E P1 P2 P3 S

Table 2.5: Box constrained ESA MGADSM problems. E: Earth, V: Venus J: Jupiter, S: Saturn, Me: Mercury, M: Mars, 67P: Comet 67P/Churyumov-Gerasimenko, Pi: generic planet chosen in the set $\{E, V, M, J\}$

All tests, if not otherwise stated, have been performed with the following stopping criteria:

- Multistart performed 1000 steps, with uniformly generated starting points;
- at each Multistart step, MBH was executed until no improvement was observed in the last $N = 500$ iterations

We report our results using diagrams similar to performance profiles ([63]), using the current performance function:

$$f(t) = \frac{|\{i \in 1..\texttt{NRuns} : f_i \leq f^\star \times (1 + t/100)\}|}{\texttt{NRuns}}$$

$f(t)$ represents the percentage of times (out of 1000 independent trials) in which the algorithm obtained a final value which was within $t\%$ of the currently known putative optimum.

In practice, at $t = 0$ we can read the percentage of times (if any) the method obtained the currently known global minimum, at $t = 100$ the fraction of runs giving a value which is at most twice the optimum, i.e. no more than 100% worse, and so on. This way it is quite easy to read from the figure which algorithm gave the best approximation to the optimum and which was capable of producing a larger quantity of good results. These

graphical representations are closely related to performance profiles, whereas here we compare different independent runs of some algorithms on a single problem, while usual performance profiles show the behavior of single runs of different methods on different test problems.

Results for the Tandem mission

As principal test-bed, we choose the Tandem mission, a box constrained problem with 18 variables. At the ESA ACT web site, 24 instances of box constrained missions to Saturn are proposed, differing on the particular sequence of swings-by performed. In the following figures we will represent computational profiles in particular for what concerns the mission with highest estimate of the global maximum, i.e. mission 6 (starting with Earth, with 3 swings-by at Venus, Earth and Earth again). This problem is formulated as a maximization one, as the objective is a function of the final mass of the aircraft.

The first trials we performed were devoted to understand which kind of perturbation was "optimal" during the execution of MBH. In particular, as there seemed to be some evidence that some variables in the problem like, e.g., starting times and times of flight, were in some sense easier to choose than other ones, or, at least, that, once well chosen, they were quite stable, we tried to check whether perturbations involving only a few variables at a time were more successful than perturbations in which every variable is randomly displaced. We choose the following characteristic parameters for our experiments:

- at each step of MBH, the current solution was perturbed in the following way: for algorithm labeled `MBH1PPertSome` between 1 and 4 coordinates were randomly chosen and uniformly perturbed in an interval of radius equal to 5% of the box containing the variable; for algorithm labeled `MBH1PPertAll` *every* coordinate was uniformly perturbed in an interval whose radius is 5% of the box.
- numerical derivatives were computed using a parameter $h = 10^{-5}$ in formulae (2.15)-(2.16).
- A single phase of local optimization was performed

In Figure 2.13 we report the results obtained running the two versions of this method, with the graphical representation introduced above.

It is quite evident from Figure 2.13 that perturbation of all coordinates is preferable: although both methods find similar global optima, the method based on the perturbation of all variables is significantly more robust (the corresponding curve is significantly "higher" than that of the competing algorithm).

The second set of experiments was aimed at checking the efficiency of our two-phase approach versus the single phase one. During the first-phase we let $h = 10^{-2}$ in the formula for numeric derivatives. When the local optimization method, SNOPT, called for stopping, we started a second optimization with the usual parameter $h = 10^{-5}$.

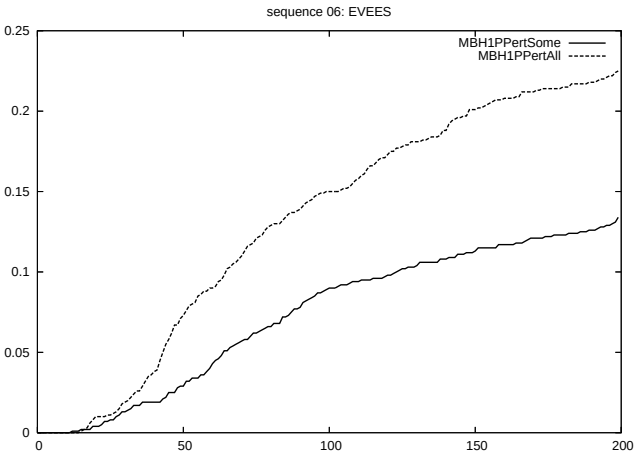


Figure 2.13: Comparison of two different perturbations: all variables (dotted line) vs. just a few ones (solid line).

Apart from this, we maintained the other parameters unchanged. In Figure 2.14 we report the comparison between one- and two-phase optimization.

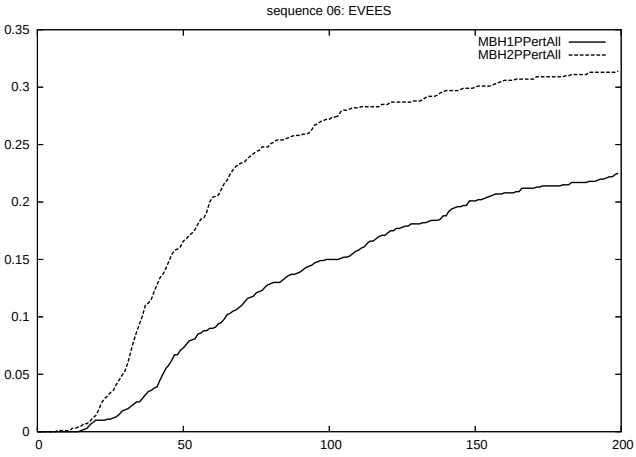


Figure 2.14: Comparison between 1– (solid line) and 2–Phase (dotted line) algorithms

Again it is evident how using two phases is extremely beneficial both in terms of precision and of robustness.

A final experiment was carried on in order to check whether MBH was indeed useful: we counted, for the 1000 experiments made, the total number of two–phase local searches performed, which resulted to be 950 046. We then ran the same number of (two-phase) Multistart iterations and checked the obtained result. In Figure 2.15 we report the comparison between MBH and Multistart; in the figure, for what concerns Multistart, we

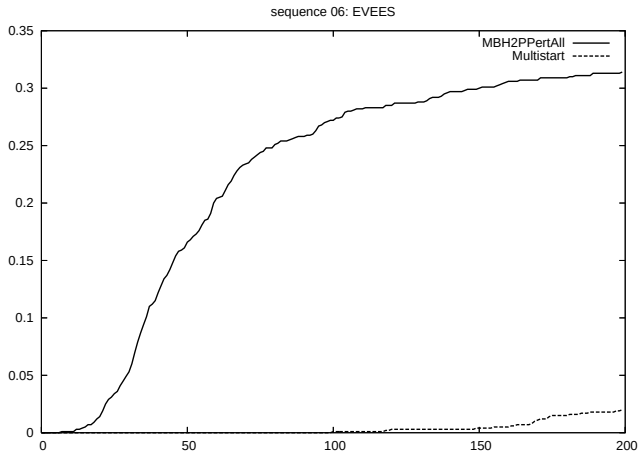


Figure 2.15: Comparison between Multistart and MBH

choose the 1000 best results and compared them with MBH – it is clear that this way the behavior of Multistart is artificially much improved; nonetheless the superiority of MBH is striking. This fact might lead us to conjecture that, similarly to problems in molecular conformation, also space trajectory optimization possesses a "funnel" structure, in which minima are clustered together. The fact that trajectory planning problems might possess a funnel structure apparently was never noticed or conjectured in the literature.

It is worth mentioning that repeating the same kind of analysis on all the 24 tests for the Tandem mission gave roughly the same behavior.

Results for other box-constrained missions

We also ran experiments on the other box-constrained tests proposed by ESA ACT, namely those related to Rosetta, Messenger and Cassini missions. Again, the same kind of algorithm was particularly efficient in all of those tests. It can be easily seen from the ESA ACT web site that, as of the time of writing, we were record holders for all of the box-constrained tests available.

In particular, we consider particularly interesting the fact of having been able, for what concerns the Messenger mission, to discover a competitive solution which corresponds to starting the mission years before the starting date of previously known solutions (as well as of the real space mission).

We would like to remark that most of the novel putative global optima we found are truly new solutions, i.e. they cannot be considered as refinement of previously known ones. As an example, we plot in the following figures a trajectory assumed to be optimal for Rosetta mission on April 2008, with objective function (representing total mission variation in velocity) equal to 1.417km/s, and the one we found (and later improved) in May 2008, with objective 1.3678. Although the variation in the objective is not

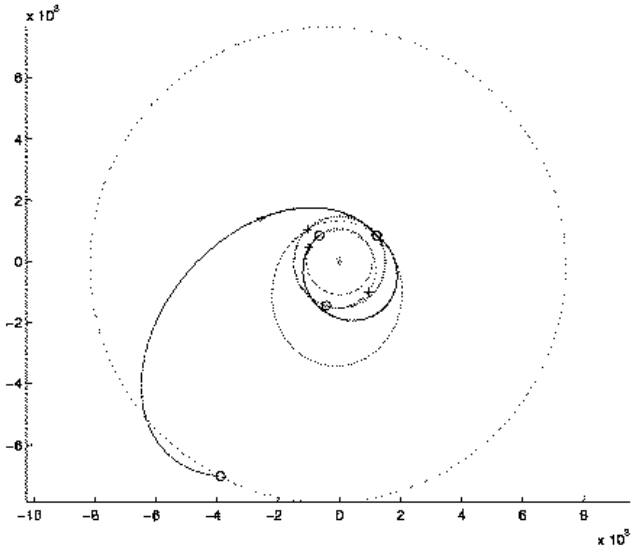


Figure 2.16: Rosetta mission, $\Delta_V = 1.417$

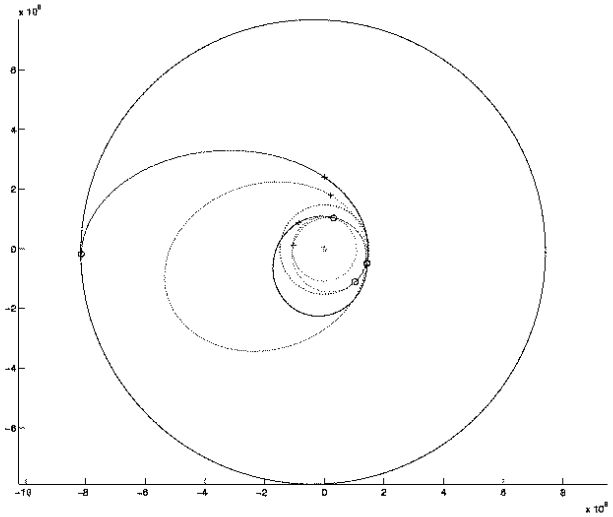


Figure 2.17: Rosetta mission, $\Delta_V = 1.3678$

particularly impressive, the trajectories widely differ.

For more details on the results, the interest reader can refer to the full paper[J19].

2.4 Conclusions

In this chapter, after a brief introduction on global optimization, I propose some ideas to interpret certain global optimization strategies as the result of introducing some form of information (local and global) in very simple stochastic methods. Then, I use some of my contributions, namely, circle packing and space trajectory design to give a more concrete example of how such information can be introduced on specific application problems. For both problems, the proposed methods allowed to determine new putative optima, and for a circle packing problem to win an international optimization competition.

3 Global optimization for membrane gas separation processes

Membrane gas separation by means of synthetic polymeric membranes is a well-established technology for several industrial applications such as production of nitrogen from air, hydrogen recovery from ammonia plants and refineries, natural and bio gas treatment, and vapor recovery from vent and process gas streams. Intensive R&D efforts are underway to further expand the number of applications of gas separation membranes [79]. Membrane separation processes are based on two pillars, membrane separation performance, derived from intrinsic separation performance (a classic subject of research in materials science) and process design, namely the choice of operating conditions and interconnections between the selected equipment. Even if the focus of our work is on the second pillar, it is necessary to open a parenthesis to explain (in a very simplified way) how membrane separation works. Membranes for gas separation are constituted by specific materials that have a selective permeation capacity with respect to different gases, that means that if a mixture of different gases enters the membrane (the so called *feed*), some components would pass through it like a pipe (forming the *retentate* or *retentated flow*) and some others will drop along the material, because for them the material is porous (determining the *permeate* or *permeated flow*). Just as an illustrative example (see Figure 3.1), let us consider that a feed containing natural gas (methane, CH_4) that must be "cleaned" from impurities (at extraction natural gas is never composed only of CH_4, but can contains other gases in different proportion, such as CO_2). In the figure the CH_4 is represented by the small white circles and the dark one represent the impurities (CO_2). Thanks to a difference of pressure, the flow entering the membrane is split in two flows (retentate and permeate), each of them with a different composition w.r.t. the initial gas: the retentate will be rich in CH_4 and the permate will be rich in CO_2.

It must be observed that in realty membrane separation is never perfect, this means that the retentated flow will still contain some impurities (some small dark circles), even if their proportion will be smaller than in the original gas mixture. The quality of the separation depends on different elements: material properties, membrane area, applied pressure. Therefore, several separation stages (membranes) are often necessary when applications demand high levels of recovery[1] and purity[2] of one or several components, or when the feed is poor in the component(s) to be recovered, or both. Multiple process

[1] the level of recovery is calculated as the ratio between the final product and the quantity of such component in the input gas

[2] purity represents the percentage of the wanted gas in the final product and it is a complementary measure of the residual presence of "unwanted" gases

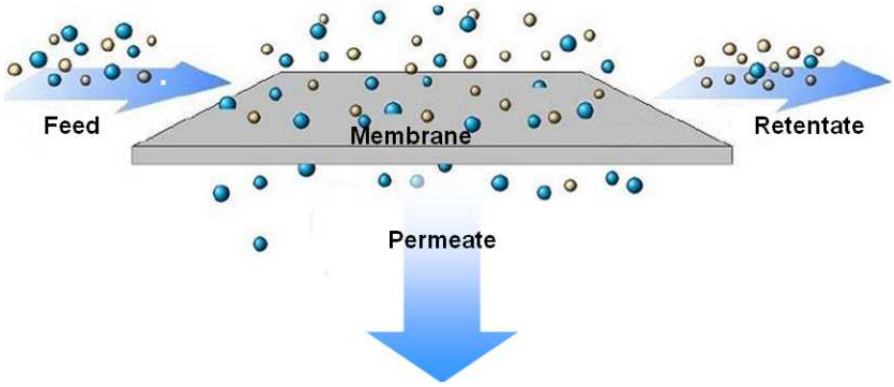


Figure 3.1: Membrane schematic functioning

configurations are possible when considering a multi-stage membrane separation i.e. number of membrane stages, membrane surface of each stage, upstream and downstream pressure of each stage and stream connections among stages, and thus finding the optimal design for the selected membrane(s) becomes the main objective of process design. In Figure 3.2, an example of a separation process for natural gas is reported. For each flow, the gas composition and pressure are reported. On each stage the total area of the membrane is reported. We can observe that starting from a 73% of CH_4 (and a 19% of CO_2), we obtain a product with 88.63% of CH_4 and only 2% of CO_2.

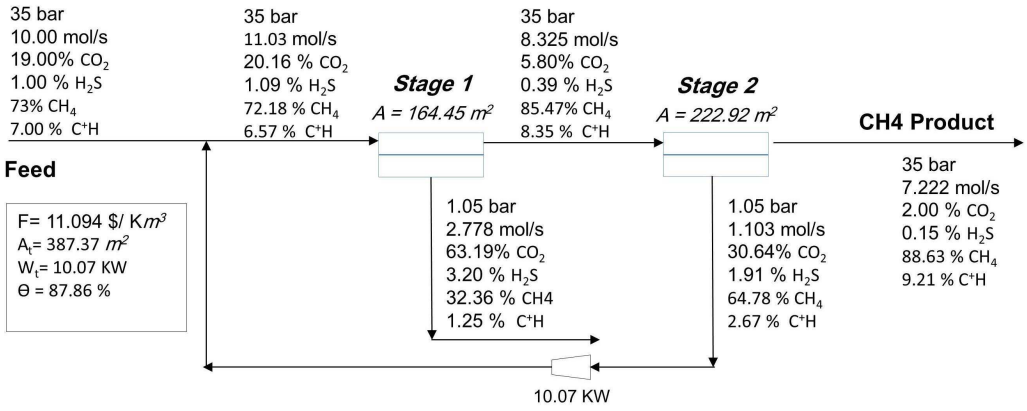


Figure 3.2: Example of a membrane gas separation process

The design of optimal membrane gas separation processes becomes more and more complex as the number of separation stages increases [154], making the traditional simulation-based approaches for process design tedious and time-consuming and often leading to solutions far from the optimal one, due to the evaluation of a limited number of process configurations.

The optimization approach for process design aims to explore (ideally) the full set of

meaningful configurations in order to ensure that the process operates under the most favorable conditions.

Membrane system design belongs to the broader family of process design problems ([71],[45]). In such field, in order to represent in an aggregate manner all possible process configurations in terms of allowed units (types and numbers) and connections between them, the superstructure representation/paradigm was introduced in the early 90s ([71]). Figure 3.3 is an illustrative example of superstructure representation with 3 membranes (stages). The dotted lines represent all possible connections. For each stage, both retentate and permeate can be split (small triangles in Figure 3.2) and sent directly to the output and/or to any stage, including itself (self-loop/self-recycling).

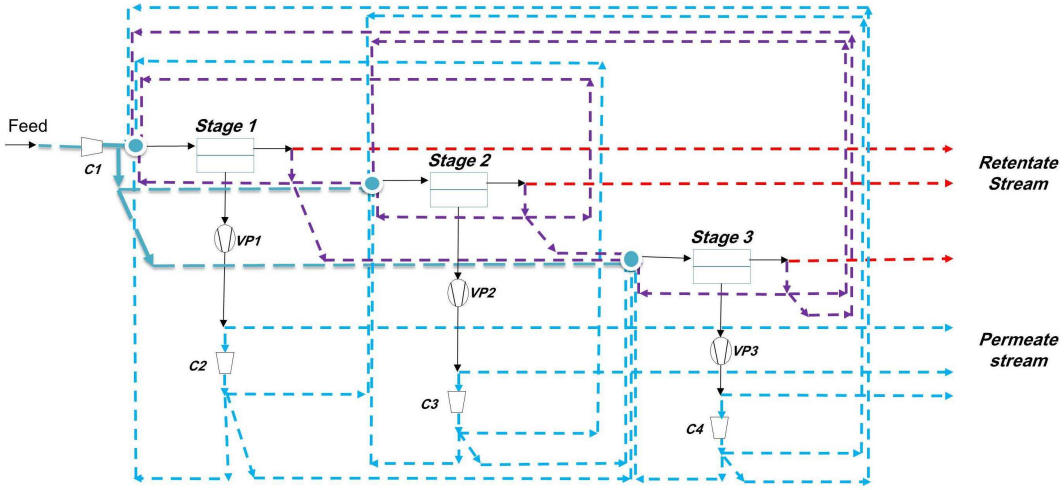


Figure 3.3: Example of a membrane gas separation process

The main advantage of this approach is to have a single compact representation of the system and to drive the transcription of all the feasible configurations in a mathematical programming model. Superstructure representation has been introduced in membrane systems design by the landmark works [145, 146], and used in several successive works ([159],[150],[138],[8]). In each of these papers, different assumptions are made on the superstructure to reduce the feasible space of the resulting membrane designs (see Table 3.1). These simplifying assumptions are aimed at removing configurations that are not physically meaningfull (for example full self-loop of the retenate or the permeate), or supposedly not effective from the practical point of view. Another advantage of these assumptions is that the search space of the resulting mathematical programming model is reduced in size or made simpler to explore. For example the absence of partial self-loops reduces the instability of the underlying non-linear local optimization problem, a reduced/fixed number and position of membranes reduces the number of integer variables necessary to represent the superstructure, etc.

Starting from a given superstructure, it is possible to derive a mathematical programming model of the system design process, that can in some cases involves discrete vari-

ables leading to a non-convex MINLP ([146]), whereas in some other case involves only continuous variables leading to a NLP [14]. The resulting NLP (describing the complete problem or the one that can obtained by fixing discrete variables in the case of MINLP models) is a non-convex problem where non-linearities are mainly in form of bilinear equality terms plus additional higher order equalities. Therefore, even the feasibility problem underlying the membrane system design is a challenging global optimization problem.

The objective function used for the optimization is typically expressed in terms of the total process costs after accounting for the main cost elements of the process (membranes, compressors, etc. both in terms of capital and operational expenditure), or as specific costs when the total costs are expressed by unit of the recovered component, or as a revenue function when a market price is attributed to the recovered gas. Depending on the specific separation conditions, adding an additional separation stage can be cost-effective, however the general trend is an increase in the total process costs as the number of stages exceeds three/four stages. This is mainly because of the costs arising from the need of additional compression/vacuum equipment for each new membrane stage. The cost of an eventual increase in total membrane surface must also not be neglected. For most of the gas separations of industrial relevance reported in the literature, the number of stages typically varies between one and three [68, 144, 159, 150, 138, 77], and in some cases up to four stages are considered [14, 146, 145]. In the following, a short literature review is reported.

3.1 A short literature review

Optimization approaches for multi-stage membrane separations have been reported in scientific literature mainly for natural gas and biogas applications[146, 150, 138, 144, 145], and for CO_2 capture applications [14, 77, 176]. Uppaluri et al.[159, 160] reported the application of an optimization approach to enriched oxygen production, nitrogen enriched air, and to hydrogen recovery from synthesis gas and refinery streams. Most of these works have been based on the optimization of the mathematical model derived by a process superstructure. There is a trade-off between increasing the size and complexity of the superstructure, which allows for the evaluation of more process configurations, and dealing with more complex problems from a mathematical point of view. Varying process assumptions defining the possible configurations included in the superstructure as well as different optimization methods have been considered in previous works. Table 3.1 presents a non-exhaustive list of the most recent studies summarizing the main process assumptions of the system and modeling approach.

Qi et al.[146] proposed a membrane system configuration to separate multicomponent gas mixtures, based on an approximate permeator model and a MINLP. A superstructure approach is presented, allowing the simultaneous optimization of the process configuration and operating conditions of a very large number of possible process alternatives. Different multi-stage membrane configurations were presented. Process layout and area are the main optimization variables. Feed side pressure, equal for all stages, was also

considered as an optimization variable while permeate pressure was free to vary for each stage respecting a fixed value of the outlet CO_2-rich permeate of around 1 bar and restricted to pressures no lower than this value. Process layouts with variable feed-side pressure among membrane stages and vacuum operation on the permeate side were thus not included in the superstructure.

In the work by Uppaluri et al. [159], the cost minimization of the membrane separation problem is also modeled as a MINLP problem with a superstructure based approach. The optimization strategy is a simulated annealing algorithm with an exploration of the neighborhood based on structure and stream moves. The structure moves change the structure of the network (membrane areas, number of stages, flow configurations), while the stream moves only change the flow distribution. Therefore, the exploration is both in terms of continuous and discrete variables. For each new generated point, once integer variables are fixed, the nonlinear equations describing the system behavior are imposed through a standard nonlinear equation solver. Then, this point is simulated, the cost function is computed, and the corresponding solution is accepted or rejected. Performance constraints are not directly inserted in the model, but enforced through a penalty term, therefore, solutions found by the optimization phase could be unfeasible. Upstream and downstream pressures are allowed to vary but are the same for each membrane stage. The use of vacuum pressure on the permeate side was considered in the evaluated configurations.

Yuan et al. [176], formulated a multi-objective optimization model to simultaneously minimize energy consumption and membrane area of six single and dual-stage membrane process designs for carbon capture from a CO_2-N_2 binary mixture. The optimization model is solved by means of a nondominated sorting genetic algorithm (NSGA-II) available in MATLAB's Global Optimization Toolbox. Operating upstream and downstream pressures, temperatures of inlet streams to compressors, vacuum pumps, and expanders, and intermediate composition have been considered as decision variables in the optimization which sought to minimize energy consumption and total membrane surface simultaneously. No cost modeling was performed for the process based on an ideal-N_2 selective metallic membrane.

Scholz et al. [150] presented a cost minimization for multi-stage configurations, modeled as a MINLP problem in a superstructure approach for biogas upgrading, and solved by means of the Branch-And-Reduce Optimization Navigator (BARON) solver. The feed was modeled as a CO_2-CH_4 binary mixture. A practical maximum number of three stages was defined for the study although the method presented is not limited to any number of stages. Upstream and downstream pressures are allowed to vary and can be different for each membrane stage. The use of compressors, vacuum pumps and pressure reduction valves were considered in the superstructure. The need to produce the upgraded gas at a pipeline pressure of 16 bar favored the use of feed compressing over permeate vacuum, and optimal process configuration remained the same for the required upgrading conditions when comparing membranes with a CO_2 permeance of 60 GPU and CO_2/CH_4 selectivities of 20 and 60. Interestingly, the resulting optimal configuration matches the configuration of a three stage process previously patented by Evonik industries [158]. Furthermore, the optimal CO_2/CH_4 selectivity for the process was also

studied by considering the upper bound correlation of Robeson within the optimization problem. The optimized process configuration based on the optimal ideal membranes consisted of two stages, instead of the three necessary for the optimal configuration when commercial membranes are used, however the authors explain that the impact of the optimal membranes on overall profitability might not be enough to support the commercial development of new membranes for this application [150].

The same optimization approach was presented by Ohs et al. [138] for the removal of nitrogen from natural gas. The superstructure was limited to two stages and the effect of nitrogen selective vs. methane selective membranes on process configuration was evaluated. The use of both kinds of membranes in the same two-stage configuration led to a lower process costs compared to the cost obtained by using only one kind of any of the two types of membranes. It was shown that new, more selective membranes would allow a noticeable reduction of process costs.

Arias et al.[14] presented recently a NLP optimization approach of a superstructure of process configurations for CO_2 capture from coal-fired power plants at recovery-purity targets ranging from 90 to 98%. The feed was modeled as a binary CO_2-N_2 mixture. The process layout and operating variables are considered as decision variables, as well as the optimal number of stages. Upstream pressure is allowed to vary but is the same for each stage and downstream pressure is fixed for all stages. The use of vacuum pressure on the permeate side was not considered in the evaluated configurations. An analysis of the equations involved in their model, showed that they could get rid of the discrete variables describing the configuration, allowing thus a NLP formulation instead of a MINLP. The NLP problem has been solved by means of CONOPT 3.0, using some heuristic to help generate a good starting solution with respect to feasibility. Results presented illustrate how the optimal number of stages, and process configuration, depend of the imposed purity-recovery constraints. The optimal two-stage configuration obtained agreed with that typically used in most capture studies while interesting configurations were proposed for processes with three and four membrane stages.

Gabrielli et al. [77], presented a multi-objective optimization model which was defined and solved by means of a genetic algorithm to minimize the specific energy consumption and required membrane area of CO_2 capture from a binary CO_2-N_2 flue gas mixture. Four fixed single and dual-stage process configurations were first analyzed by a parametric analysis and then the optimal configuration was evaluated for up to six membrane stages. Process layout, membrane selectivity according to Robeson's upper bound, and operating variables are considered as decision variables. Upstream pressure is fixed and equal to the feed pressure at atmospheric conditions, and permeate pressure is varied between 0.01 and 1 bar independently for each stage. A global minimum in terms of energy consumption was identified for the enricher configuration with recycle at three membrane stages, and across the selectivity range evaluated (20-60).

Aliaga-Vicente et al. [8] proposed a MINLP approach for two superstructures dealing with CO_2 separation from CO_2/CH_4 mixtures for natural gas sweetening and enhanced oil recovery. They employed the SBB algorithm which is a combination of the standard branch and bound method and CONOPT in the GAMS software. The superstructure approach is based on the evaluation of separation cascades as presented by Agrawal et al.

[4] and included novel aspects such as the consideration of an expander for the exiting retentate stream, lateral extraction streams and the consideration of temperature changes during membrane separations and permeate compression by means of correlations obtained in Aspen Hysys. The process layout and operating variables are considered as decision variables, as well as the optimal number of stages. Upstream pressure is allowed to vary but the use of vacuum pressure on the permeate side was not considered in the evaluated configurations. It is explained by the authors that the approach presented does not guarantee a global optimal solution because of the nonlinearity and non-convexity of the optimization problem.

Table 3.1: Recent publications on multistage gas membrane separation optimization

Industrial application	Only binary Feed	Modeling approach	Assumptions on the system: Upstream pressure	Downstream pressure	Max# stages	Permeab./ Select.	Ref.
N_2 removal from natural gas	YES	Superstructure No self recycling	Fixed	Free and uniform	2	Fixed Function	[138]
Air separation H_2 recovery from synthesis gas and refinery streams	NO	Superstructure	Free and uniform	Free and uniform	3	Fixed	[160]
Biogas upgrading	YES	Superstructure No self recycling	Free and indep.	Free and indep.	3	Fixed Function	[150]
CO_2 capture from coal-fired power plants	YES	Superstructure with restricted interconn. No feed split	Free and uniform	Fixed	4	Fixed	[14]
CO_2 capture from coal-fired power plants	YES	Fixed configurations	Fixed	Free and uniform	6	Fixed	[77]
CO_2 capture from coal-fired power plants	YES	Fixed configurations	Free and indep.	Free and indep.	2	Fixed	[176]
Natural gas sweetening and enhanced oil recovery	YES	2 superstructures with restricted interconn. no feed split	Free and uniform	Free and uniform	4	Fixed	[8]
CO_2 capture from industrial gas	NO	Superstructures with restricted interconn.	Free and indep.	Free and indep.	3	Fixed	[67]
CO_2 capture from coal-fired power plant	YES	Superstructures with restricted interconn.	Free and indep.	Free and indep.	3	Fixed	[151]
Natural gas sweetening and enhanced oil recovery	NO	Superstructure	Free and uniform	Free and indep.	4	Fixed	[146]
CO_2 capture from blast furnace gas	NO	Superstructure	Free and uniform	Free and indep.	4	Fixed	[J1]

In our study ([J1]), we present a global optimization approach solving a NLP formulation of the optimization of membrane architectures and validate it on CO_2 capture from blast furnace gas multicomponent mixtures. General process superstructures with up to 4 stages are considered, and feed compression, vacuum on the permeate side, and mixed operation (using both) are all included in the superstructure (see Figure 3.3). Interconnection possibilities among process units have been kept as numerous as possible to increase the number of possible process configurations. Upstream and downstream pressure, process layout and membrane surface of each stage are the decision variables. Furthermore, downstream pressure is allowed to vary independently for each membrane stage thus allowing an independent pressure ratio for each stage. To our knowledge, no superstructure-optimization approach for membrane separation has been presented in the scientific literature with these characteristics (multicomponent feed, exhaustive connection layout including recycling loops, variable pressure ratio for each stage with compression and/or vacuum possibilities).

3.2 Mathematical programming model

In the following we present the mathematical optimization model we used, such model is based on some quite common assumptions used in the literature (the interested reader can refer to [J1] for details), it refers to a generic number of gas components and number of stages defined by as follows:

- $\mathcal{S}$: set of stages (membranes)
- C: set of gas components
- $\mathcal{C}$: set of discretization cells in a membrane

The optimization variables are described one by one in Tables 3.2, 3.3, 3.4. Here, we explain the convention we used to name them. The model is divided into three levels, with increasing level of details (see Figure 3.4): the overall system (with one given input, the Feed, and two outputs: the retentate and the permeate), the single stage constituted by a membrane $s \in \mathcal{S}$ (one input and two output flows, retentate and permeate) and, finally, the cell level, describing the behavior of the gas flow into a single cell $i \in \mathcal{C}$ of a given membrane $s \in \mathcal{S}$. Cells do not correspond to real, but are the result of the discretization process necessary to model a single membrane separation process (see equation (3.8)).

To reduce the number of different variable names, we decided to use, with a slight abuse of notation, the same letter to represent all the flows at a given level: F for the system level, f for the membrane level and g for the cell level. We use a similar convention for flow composition: X for the system level, x for the membrane level and y for the cell one.

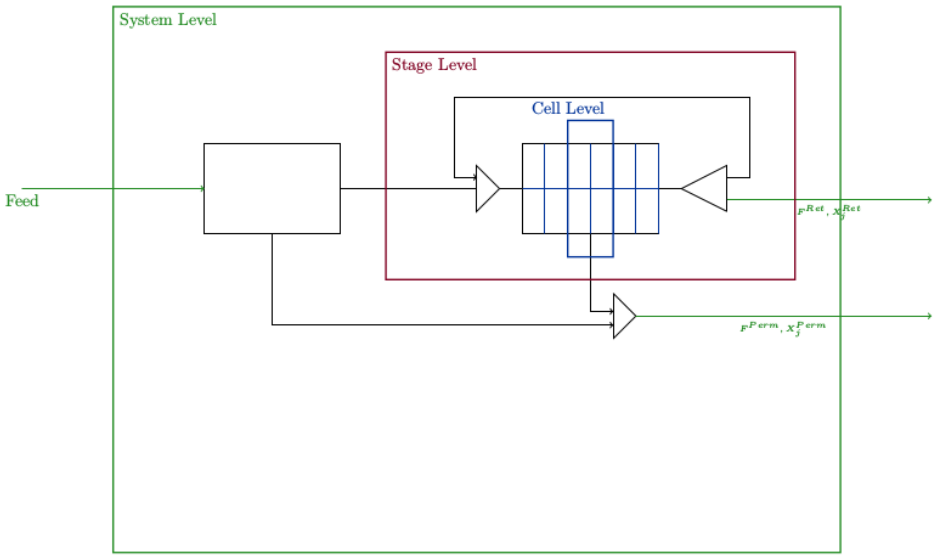


Figure 3.4: Variables structure

F^{Perm}	Permeate flowrate	[mol/s]
F^{Ret}	Retentate flowrate	[mol/s]
X_j^{Perm}	Fraction of component j in the permeate	[Dimensionless]
X_j^{Ret}	Fraction of component j in the retentate	[Dimensionless]

Table 3.2: Overal system Variables

We distinguishing among permeate and retentate with the use of the superscripts *perm* and *ret*, respectively. Subscripts are used to distinguish between different membranes, cells and, when necessary, gas components. For example f_s^{ret} represents the retentate of membrane s. At membrane level both retentate and permeate can be split to be distributed as an input to other membrane and/or sent out of the system. To represent this possibility, additional variables are needed, that we introduce adding an additional superscript when necessary:

- split: it represents the flowrate that goes from one stage to another, e.g. $f_{s,s1}^{perm,split}$ represents the quantity of permeated flow of membrane s that is redirected to membrane $s1$.
- out: it expresses the flowrate that goes from one stage to the system output, e.g. $f_s^{perm,out}$ expresses the quantity of permeated flow of membrane s that is directed to the system output.

$\mathrm{Acell_s}$	Cells' area	[m2]
$\mathrm{P^{up}}$	Up stream pressure (same for all) membranes	[bar]
$\mathrm{P_s^{down}}$	Down stream pressure	[bar]
$\mathrm{f_s}$	Overall feed flowrate	[mol/s]
$\mathrm{f_s^{split}}$	Fresh feed flow rate entering the membrane	[mol/s]
$\mathrm{f_s^{perm}}$	Total permeate flowrate	[mol/s]
$\mathrm{f_s^{ret}}$	Total retentate flowrate	[mol/s]
$\mathrm{x_{s,j}}$	Fraction of component j in the feed	[Dimensionless]
$\mathrm{x_{s,j}^{perm}}$	Fraction of component j of the permeate	[Dimensionless]
$\mathrm{x_{s,j}^{ret}}$	Fraction of component j of the retentate	[Dimensionless]
	Membrane connection variables $\forall s \in S$	
$\mathrm{f_s^{ret,out}}$	Retentate flowrate going out the system	[mol/s]
$\mathrm{f_s^{perm,out}}$	Permeate flowrate going out the system	[mol/s]
$\mathrm{f_{s,s_1}^{perm,split}}$	Permeate flowrate entering into membrane s_1	[mol/s]
$\mathrm{f_{s,s_1}^{ret,split}}$	Retentate flowrate entering into membrane s_1	[mol/s]

Table 3.3: Membrane variables $\forall s \in S$

$\mathrm{g_{s,i}}$	Feed flowrate	[mol/s]
$\mathrm{g_{s,i}^{perm}}$	Permeate flowrate	[mol/s]
$\mathrm{g_{s,i}^{ret}}$	Retentate flowrate	[mol/s]
$\mathrm{y_{s,i,j}}$	Fraction of component j of the feed	[Dimensionless]
$\mathrm{y_{s,i,j}^{perm}}$	Fraction of component j of the permeate	[Dimensionless]
$\mathrm{y_{s,i,j}^{ret}}$	Fraction of component j of the retentate	[Dimensionless]

Table 3.4: Single cell variables $\forall s \in S, i \in C$

The model constraints can be considered divided in different families: flow conservation constraints that must be valid at each level of the model and that must hold both for total flow that for each single component flow; connecting constraints allowing to connect (coherently) flows at different levels and between different membranes; coherence in the total composition of each flow; and finally the constraints describing separation (see (3.8)). We can now introduce them, one by one, considering different families and model levels.

Flow conservation constraints at the level of the overall system:

$$Feed = F^{Ret} + F^{Perm} \tag{3.1}$$

$$F^{Ret} = \sum_{s \in \mathcal{S}} f_s^{ret,out} \tag{3.2}$$

$$F^{Perm} = \sum_{s \in \mathcal{S}} f_s^{perm,out} \tag{3.3}$$

Feed	Flow rate input to the system	[mol/s]
P_{in}	Inlet stream pressure	[bar]
P_j	Permeance of component $j \in C$	[GPU]

Table 3.5: Parameters

Flow conservation constraints for each membrane s:

$$f_s = f_s^{ret} + f_s^{perm} \quad \forall s \in \mathcal{S} \tag{3.4}$$

and flow conservation constraints at the cell level (both for total flow and single component flow):

$$g_{s,i} = (g_{s,i}^{ret} + g_{s,i}^{perm}) \qquad \forall s \in \mathcal{S}, i \in \mathcal{C} \tag{3.5}$$

$$g_{s,i+1} = g_{s,i}^{ret} \qquad \forall s \in \mathcal{S}, i \in \mathcal{C} \tag{3.6}$$

$$y_{s,j,i+1} = y_{s,j,i}^{ret} \quad \forall s \in \mathcal{S}, j \in C, i \in \mathcal{C} \tag{3.7}$$

The following equations describe the membrane behavior:

$$g_{s,i}^{perm} y_{s,j,i}^{perm} = Acell_s P_j (P^{up} y_{s,j,i}^{ret} - P_s^{down} y_{s,j,i}^{perm}) \quad \forall s \in \mathcal{S}, j \in C, i \in \mathcal{C} \tag{3.8}$$

This is one of the core elements of the model, and it requires a brief explanation. We observe that the permeate fraction of gas j ($y_{s,j,i}^{perm}$) depends on the cell area ($Acell_s$), on the permeability to component j (P_j), and the ratio between the applied pressures ($\frac{P^{up}}{P_s^{down}}$)[3]. Larger areas (and/or pressure ratios) are associated to an increase of the permeate of the most permeable component. Nevertheless, the equation is highly non-linear, and does not allow to obtain directly the value of each component at the exit of a single membrane (even considering a single stage) without solving the overall system of non-linear equations.

The following equations model the correlation between the membrane level and the cell level:

$$f_s = g_{s,1} \qquad \forall s \in \mathcal{S} \tag{3.9}$$

$$x_{s,j} = y_{s,j,1} \quad \forall s \in \mathcal{S}, j \in C \tag{3.10}$$

$$f_s^{ret} = g_{s,n_s}^{ret} \qquad \forall s \in \mathcal{S} \tag{3.11}$$

$$x_{s,j}^{ret} = y_{s,j,n_s}^{ret} \quad \forall s \in S, j \in C \tag{3.12}$$

$$f_s^{perm} = (\sum_{i \in \mathcal{C}} g_{s,i}^{perm}) \qquad \forall s \in \mathcal{S} \tag{3.13}$$

$$x_{s,j}^{perm} f_s^{perm} = (\sum_{i \in \mathcal{C}} y_{s,j,i}^{perm} g_{s,i}^{perm}) \quad \forall s \in \mathcal{S}, j \in C \tag{3.14}$$

[3]to be precise the driving force needed for separation is the difference between the retentate partial pressure of component j ($P^{up} y_{s,j,i}^{ret}$) and its permeate partial pressure ($P_s^{down} y_{s,j,i}^{perm}$))

The retentate flow of a membrane corresponds to the retentate of the last cell (see equations (3.11)-(3.12)), whereas the permeated is the results of the mixture of the permeated flows of all the cells (see equations (3.13)-(3.14)). Indeed, like the flow running in a pipe with several holes, the overall water loss depends on the sum of all the falling drops. The connection between the system level and the membrane level is regulated by:

$$F^{Ret} = \sum_{s \in \mathcal{S}} f_s^{ret,out} \tag{3.15}$$

$$F^{Perm} = \sum_{s \in \mathcal{S}} f_s^{perm,out} \tag{3.16}$$

At each level, it must be enforced that the fractions of the components sum up to one:

$$\sum_{j \in C} X_j^{Ret} = 1 \tag{3.17}$$

$$\sum_{j \in C} X_j^{Perm} = 1 \tag{3.18}$$

$$\sum_{j \in C} x_{s,j}^{ret} = 1 \qquad \forall s \in \mathcal{S} \tag{3.19}$$

$$\sum_{j \in C} x_{s,j}^{perm} = 1 \qquad \forall s \in \mathcal{S} \tag{3.20}$$

$$\sum_{j \in C} y_{s,j,i}^{ret} = 1 \quad \forall s \in \mathcal{S}, i \in \mathcal{C} \tag{3.21}$$

$$\sum_{j \in C} y_{s,j,i}^{perm} = 1 \quad \forall s \in \mathcal{S}, i \in \mathcal{C} \tag{3.22}$$

Finally, flow conservation constraints related to the superstructure (split coherence) are introduced:

$$f_s^{ret} = \sum_{s_1 \in \mathcal{S}} f_{s,s1}^{ret,split} + f_s^{ret,out} \quad \forall s \in \mathcal{S} \tag{3.23}$$

$$f_s^{perm} = \sum_{s_1 \in \mathcal{S}} f_{s,s1}^{split,perm} + f_s^{perm,out} \quad \forall s \in \mathcal{S} \tag{3.24}$$

$$f_s = \sum_{s1 \in \mathcal{S}} f_{s1,s}^{ret,split} + f_{s1,s}^{perm,split} + f_s^{split} \quad \forall s \in \mathcal{S} \tag{3.25}$$

$$Feed = \sum_{s \in \mathcal{S}} f_s^{split} \tag{3.26}$$

Depending on the case study, some performance constraints can be added on purity (percentage presence of some components in the product) or on the recovery (quantity of final product with respect to its availability on the input feed). Furthermore, alternative formulations are possible, for example introducing fractions of split flows.

All variables are bounded to be in a box where the lower bound is nonnegative, and the upper bound can be derived by the physical meaning of the variable. Finally, a maximum recycling ratio is imposed to the splits of any stage towards itself (self-loops) to avoid solutions that would not lead to physically stable configurations (full self-loops).

3.2.1 Objective function: specific separation cost

We performed tests on two different case-studies, the first one was used to validate our mathematical model and optimization procedure (comparing with results presented in [146]) and the second one to show the full potential of our method on an industrial application. For the first group of tests, we implemented the objective function proposed in [146]. For the second one, we used the objective function proposed in the literature for this specific application (see [147]). Here, we describe briefly this second objective function that represent the total annual separation costs, i.e. the overall cost of separating CO_2 divided by the overall CO_2 production. The total cost consider both capital and operational expenditure (CAPEX and OPEX in what follows):

$$SC_{CO_2} = (CAPEX \cdot a + OPEX)/M_{CO_2\ per\ year} \tag{3.27}$$

Capital cost includes membrane area and frame, compressors and vacuum pumps. The total capital expenditure (CAPEX) is calculated from the equipment cost investment supposing an indirect cost factor of 80% (parameter a). Operational costs include electricity related to compression and vacuum equipment, membrane replacement, operation and maintenance cost. A detailed description (and explanation of the parameters' values) is out of the scope of this manuscript, the interested reader can find it in our journal paper [J1]. Nevertheless, we report here all the terms (Table 3.6) and parameters (Table 3.7) used to derive the objective.

Compressors/Vacuum pumps powers calculations	
$W_{cp_f} = \frac{Feed}{\eta} \cdot \frac{\gamma \cdot R \cdot T}{\gamma - 1}[(\frac{P^{up}}{P_{in}})^{\frac{(\gamma-1)}{\gamma}} - 1]$	Feed compression
$W_{cp_s} = \frac{f_s^{perm} - f_s^{perm,out}}{\eta} \cdot \frac{\gamma \cdot R \cdot T}{\gamma - 1}[(\frac{P^{up}}{P_{in}})^{\frac{(\gamma-1)}{\gamma}} - 1]$	Retentated compression
$W_{vp_s} = \frac{f_s^{perm}}{\lambda} \cdot \frac{\gamma \cdot R \cdot T}{\gamma - 1}[(\frac{P_{in}}{P_s^{down}})^{\frac{(\gamma-1)}{\gamma}} - 1]$	Vacuum pump
$W_{tot} = \frac{W_{cp_f} + \sum_{s \in \mathcal{S}}(W_{cp_s} + W_{vp_s})}{\phi}$	Total energy consumption
Equipment costs	
$I_{m_s} = A_{m_s} \cdot K_m$	Membrane cost
$I_{mf_s} = (A_{m_s}/2000)^{0.7} \cdot K_{mf} \cdot (p^{up}/55)^{0.875}$	Membrane frame cost
$I_{cc_s} = C_{cc} \cdot (W_{cp_s}/10^6)^{0.7} \cdot MF_{cc} \cdot MDF_{cc} \cdot UF_{2000}$	Stage compressor cost
$I_{cc_f} = C_{cc} \cdot (W_{cp_f}/10^6)^{0.7} \cdot MF_{cc} \cdot MDF_{cc} \cdot UF_{2000}$	Feed compressor cost
$I_{vp_s} = C_{vp} \cdot (W_{vp_s}/10^3)$	Vacuum pump cost
CAPital EXpenditures	
$(I_{cc_f} + \sum_{s \in \mathcal{S}}(I_{m_s} + I_{mf_s} + I_{cc_s} + I_{vp_s})) \cdot ICF$	**CAPEX**
OPerational EXpenditures	
$C_{O\&M} = \sum_{s \in \mathcal{S}} A_{m_s} \cdot \nu \cdot K_{mr} + 0.03 \cdot CAPEX$	Operation and maintenance
$C_{en} = t_{op} \cdot W_{tot} \cdot K_{el}$	Energy
$C_{en} + C_{O\&M}$	**OPEX**
Annual separated CO_2	
$F^{Perm} \cdot X_{CO_2}^{Perm} \cdot M_{CO_2} \cdot 10^{-6} \cdot 3600 \cdot t_{op}$	$M_{CO_2\ per\ year}$

Table 3.6: Cost equations used to determine product gas separation cost

Table 3.7: Cost-related parameters

	Capital cost		
C_{cc}	Compressor factor	1×10^6	EUR/KW
C_{vp}	Vacuum pump factor	1500	EUR/KW
K_m	Unit cost of membrane module	40	EUR/m^2
K_{mf}	Base frame cost	286×10^3	EUR
MDF_{cc}	Compressor module factor	2.72	-
MF_{cc}	Compressor material factor	1.4	-
UF_{2000}	Update factor	1.42	-
ICF	Indirect cost factor	1.8	-
	Operational conditions/performances		
T	Temperature	308.15	K
R	Ideal gas constant	8.31446	$JK^{-1}mol^{-1}$
η	compressor efficiency	0.85	-
ϕ	mechanical efficiency	0.95	-
γ	gas expansion coefficient	1.36	-
λ	vacuum pump efficiency	0.85	-
	Annual cost		
ν	Membrane annual replacement rate	0.2	-
K_{mr}	maintenance costs	25	EUR/m^2
t_{op}	operations time per year	8322	h/year
K_{el}	electricity cost	0.044	EUR/kWh
a	annuity coefficient	0.0854	-
M_{CO_2}	Molar mass of CO_2	44.01	g/mol

3.3 Optimization procedure

Gas membrane separation process optimal design, as explained in the introduction, can be considered as a non-convex MINLP, or NLP, depending on some problem assumptions and modeling choices. Even restricting to a NLP formulation results in a difficult global optimization problems, therefore we decided to start our study considering some assumptions that allowed us to remove discrete variables: the number of membranes is fixed[4], the discretization of membranes has a fixed number of cells, and the total area is adjusted enlarging the cell area[5], all connections between membranes (and in any number) are allowed, and a zero flow (or a very small one) is considered as a connection to be deleted from the resulting design[6]; upstream pressure is a variable, but it is the same for all membranes and downstream pressures are free, but they are all bigger or lower than one (these hypothesis allow to determining the need for compressors and vacuum pump before starting the optimization procedure, and therefore remove the need of additional discrete variables to model their possible presence).

To tackle the underlying global optimization problem we worked first on the local convergence of the model, choosing a good formulation and selecting a local solver ([O1]) and then tailoring a Multistart-MBH strategy as in Algorithm 6 (for an introduction to the general method see Chapter 2).

Algorithm 6 `Global Optimization Algorithm`

$t = 0, f^{min} = \infty$
while $t < T$ **do**
  $x_t^*, f(x_t^*)$ =Monotonic Basin Hopping(N)
  if $f(x_t^*) < f^{min}$ **then**
    $f^{min} = f(x_t^*)$
    $x^{min} = x_t^*$
  $t = t + 1$
return x^{min}, f^{min}

With respect to the local strategy, we chose, after extensive testing, KNITRO as local solver ([34]). As for the global method, our attention concentrate mainly on the random generation/perturbation to allow a faster convergence of the local search, and also to try to converge "near by" the random generated points. Indeed, if the starting point is too far from the feasible region, the local solver can move variables largely in the attempt to satisfy constraints leading to completely different arrival regions even starting from points that are "close", therefore transforming the MBH strategy in a pure random search one.

Before briefly explaining our random generation procedure, we need to observe that

[4] to analyze configurations with different number of membranes we run separated tests

[5] all papers in the literature that use mathematical programming models use this hypothesis

[6] in some papers, as for example [146], that we used as comparison, discrete variables are used to decide is a flow is present or not, but these variables are unnecessary complicating the model.

the variables can be separated in two groups: independent and dependent, that roughly speaking, can be considered corresponding to design (membrane areas, pressures, fraction of splits, etc) and auxiliary variables (all cell variables, flow compositions, etc), respectively. If we discard additional constraints (related to purity and/or recovery) and the objective function, our mathematical programming model is the discretization of a system of differential equations that represent the functioning of the overall membrane process. Therefore, as it is represent a physical system, it can be expected that fixing the design variables (and considering a fixed feed with a given composition entering the system), the solution of such mathematical programming model is unique[7]. Our random generation technique is based on this observation and it is based on two steps:

- Generation/perturbation of design (independent) variables, taking into account easy linear equality constraints (e.g. when we calculate split flows, we force that the sum remains equal to the total flow), a similar idea seems implemented also in [14]
- Determining the dependent variables (flows between membranes, flow compositions, etc.) solving a simplified version of the model
- Using some simplified assumptions (linear membrane behavior) to initialize cell variables

We can consider this strategy as belonging to the family of two-phase optimization procedures (as we explained in Section 2.3.2), but aimed to enforce feasibility and not to improve the quality of the local optima. This procedure may seem rather obvious, but, to the best of our knowledge, no one described it in the current literature. Furthermore, computational results shows that adding such a procedure in a Multistart strategy, not only improves the convergence to local optima (in terms of computational time and number of success), but it seems also to improve the quality of the local optima found in terms of objective function.

3.3.1 Validation of the optimization procedure

In order to validate the optimized configurations obtained by our GO algorithm, we applied it to a reference case for which an optimization approach based on a MINLP model was proposed by Qi et al. [146]. For finding local solutions, authors used CONOPT2 for the NLP parts and DICOPT for the MINLP. The global optimization strategy is not explained into detail, but it seem reasonable to classify it is as Multistart-like approach ("initializing the variables at several different points, setting reasonable bounds on the variables, and adjusting the DICOPT++ options to facilitate convergence to the global optimum.").

The choice of this reference for comparison is motivated by its accredited value in the community and, not less important, by the presence of a full description of the applied

[7]there is no formal proof at the moment of this point, maybe because there is not the need of it for the process engineering community

model, the optimization objective and optimal configurations[8], which allowed us a clear and straightforward comparison.

We focus on the case study of natural gas treatment with four component mixture with continuous membrane areas (Section 3.2 of [146]). We keep the same assumptions, but we use our mathematical model as described in 3.2 for the membrane calculation, except for the cost model (annual process cost), that we keep as in [146]. Our superstructure for this case excludes the possibility of vacuum operation on the permeate side since this operation mode is not considered in [146]. Three sets of operating conditions were considered and evaluated during validation, with increasing degrees of freedom :

(i) both upstream and downstream pressures and membrane areas are fixed and only connections among membrane stages are allowed to change

(ii) upstream and downstream pressures are fixed but membrane area is free to vary among stages

(iii) both upstream and downstream pressures and membrane area are free to vary in a suitable range

For each of these three cases, the distribution of compressors can be determined a priori. Apart from these assumptions, we allow any connection between units (including self loops) and any split (in terms of number and possible fraction) of flows. Finally, each membrane process configuration resulting from the GO algorithm was compared (retentated and permeated flows and their composition of each stage, objective function value) to simulations performed at the same conditions by means of a proprietary gas permeation calculation tool developed at the LRGP (Reactions and Process Engineering laboratory): MEMSIC 6.0 [32], implemented to the Aspen Plus V8.6 environment through the CAPE-OPEN standard, with the objective of evaluating the numerical difference between our optimization algorithm and the classical simulation approaches that were used by our co-authors working at LRGP.

Numerical results

First of all, we decided to validate our mathematical programming model considering only local optimal solutions. Indeed, as expected, fixing design variables leads always to a single local optimum. We consider the optimal configurations reported in Figure 2 and Figure 4 of [146] which are presented here as Figure 3.5a and Figure 3.6a respectively. These configurations and their corresponding objective values are considered as the reference cases (see Table 3.8). The original values presented in [146] are reported in parentheses while the recalculated values using the same cost equations are shown without parentheses. The difference is likely due to some rounding up before computing the objective value. The values of the objective function calculated by the optimization algorithm are denoted as "GO". Finally the values denoted as "ASPEN" correspond to the values obtained when simulating the same process conditions in Aspen Plus v8.6.

[8] all such elements in a single paper are very rare

It can be seen that the differences between the three set of values are small, indicating that the difference in membrane modeling between the three approaches leads to small differences in the objective values (annual process cost) calculated in each case, and therefore showing that our mathematical model is comparable with that of the reference case and with that of a classical process simulation.

Case Name	2-stages	3-stages
Reference	11.422 (11.094)	11.236 (10.971)
GO	11.660	11.807
ASPEN	11.500	11.300

Table 3.8: Objective value comparison for model validation. Annual process cost expressed as USD/m^3. Values in parentheses taken from [146].

Fixed pressures optimization

After the validation phase, the optimization is performed adding step by step degrees of freedom. First of all we start keeping fixed the membrane areas and pressures as in [146], and we allow the optimization algorithm to change the configuration of the system, that is to change the proportion in splits, the feed distribution and the layout of membranes stages. In Table 3.9, the best results found by our GO algorithm, and then simulated by ASPEN are reported. For comparison the reference values from [146] are reported.

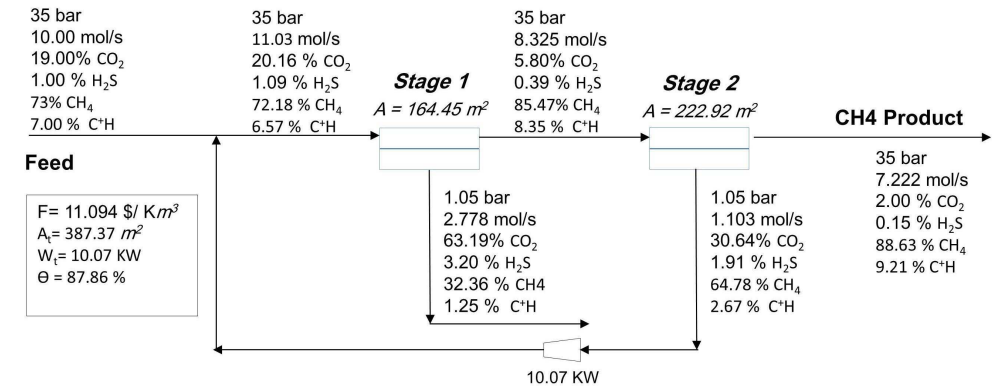
	2-stages		3-stages	
	Reference	This work	Reference	This work
GO	11.422	10.713	11.236	10.721
ASPEN	11.457	10.667	11.263	10.683

Table 3.9: Annual process cost expressed as USD/m^3. Case with fixed membrane areas and pressures. Reference values recalculated from [146]

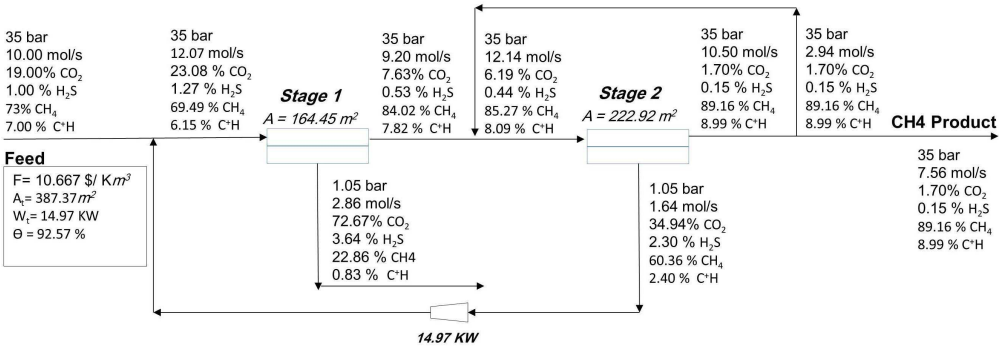
Even with the same pressures and the same membranes area, the global optimization performed here allows to achieve lower process costs when compared to the reference case, around 4 to 7%, showing its efficiency and ability to globally explore the search space. As areas are the same, the reduction in the objective is due to the reduction in the product loss term of the objective. Methane recovery is higher ($\Omega = 92.57$ vs $\Omega = 87.86$) leading to reduced costs despite the increase in total power ($W_t = 14.97 \cdot 1.64$ vs $W_t = 10.07 \cdot 1.103$).

In the two stages case, the optimal configuration is similar to the reference one, except for the inclusion of a self-recycling loop of the retentate coming from the second stage.

For the process with three membrane stages, the optimal configuration contains again a self-loop in the second stage, but other structural differences are present. Membrane area between the first and second stages is switched compared to the reference case, and it is the first stage that exhibits the larger membrane surface. Furthermore, the first stage takes as an input the permeate from the second stage. Finally, the third stage is fed by a fraction of the permeate of the first stage and not by the permeate of the second stage.

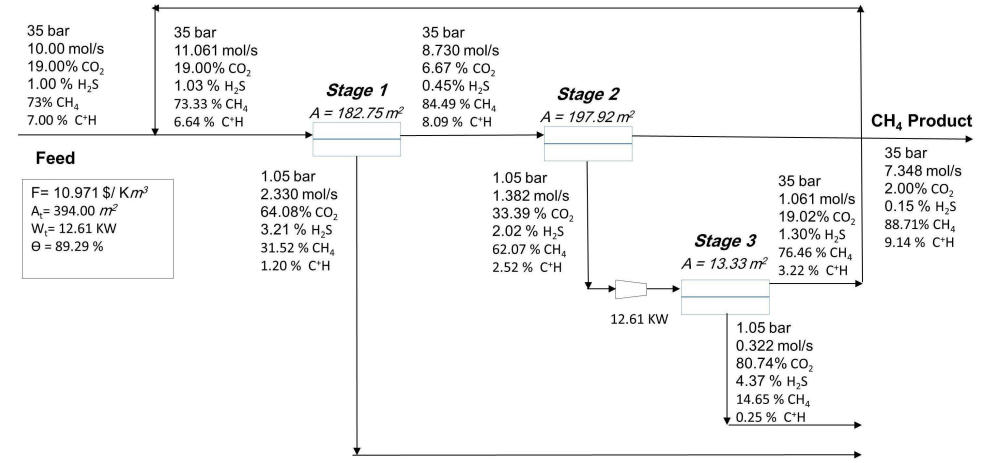


(a) Optimal 2-stage process from reference

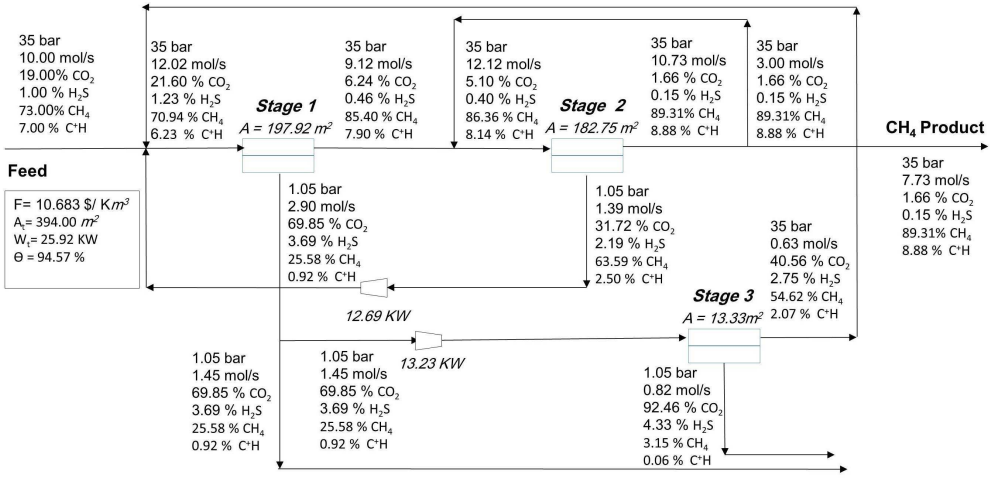


(b) Optimal 2-stage process obtained in this work

Figure 3.5: Comparison of optimization results for a two stage membrane process between the reference case [146] and the optimization approach developed in this work when both membrane area and pressure of each stage are fixed.



(a) Optimal 3-stage process from reference



(b) Optimal 3-stage process obtained in this work

Figure 3.6: Comparison of optimization results for a three stage membrane process between the reference case [146] and the optimization approach developed in this work when both membrane area and pressure of each stage are fixed.

The next step is to allow the areas of the membranes to vary, having then both the connections among stages and the membrane areas as optimization variables. We present the results only for the three stages process for the sake of brevity (and since the cost is smaller than the one with two stages). As expected, by increasing the degrees of freedom better solutions are obtained, whose objective function values are reported in Table 3.10. Once again, when compared to the optimal solution from the reference, the process configuration obtained with our approach leads to around 20% lower process cost. This is achieved by both a reduction of total membrane area and an increase in methane recovery. The optimal process calculated here and presented in Figure 3.7, has the same general configuration of the reference case presented in Figure 3.6a. No

self-recycling was included in the optimal configuration this time.

Table 3.10: Annual process cost expressed as USD/m^3. Case with fixed pressure and free area with three stages. Reference values recalculated from [146]

	Reference	This work
GO	11.236	9.115
ASPEN	11.263	9.095

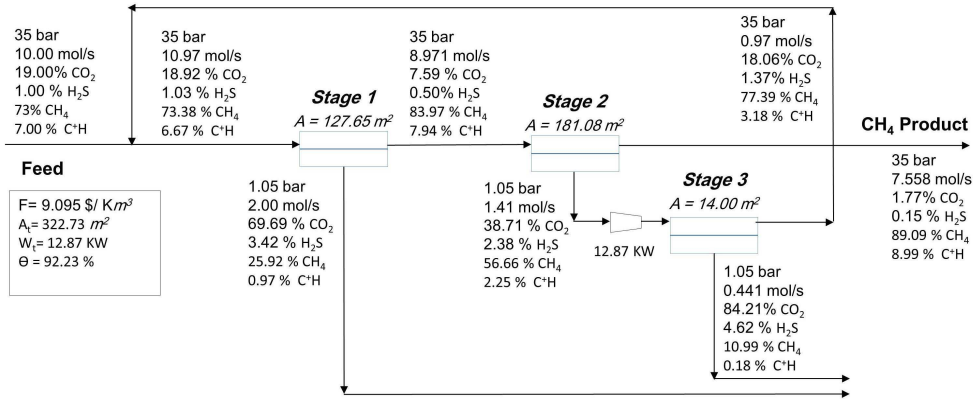


Figure 3.7: Optimal process for the case with fixed pressures and free area.

Free pressures optimization

As a further step, using the same feed conditions ($10 mol/s$ at $35 bar$), we allow both upstream and downstream pressures to vary uniformly for all the membranes stages with an upper bound of 50 bar for upstream pressure (a compressor can be added to increase pressure up to 50bar from the initial 35bar) and lower bound of 1.05 bar for downstream pressures. So, the following constraints were imposed to the system:

1. $\mathrm{P}^{\mathrm{up}} \leq 50$
2. $\mathrm{P}_{\mathrm{s}}^{\mathrm{down}} \geq 1.05$
3. $\mathrm{P}^{\mathrm{up}} \geq \mathrm{P}_{\mathrm{s}}^{\mathrm{down}} + 0.001$

The process cost reduces over 35% (Table 3.11) and the best configuration has only 2 stages (Figure 3.8) compared to the three stage process of the reference case. This is explained in part by the structure of the objective function that privileges the use of compressors over the increase of the membrane areas (this was noted also in [146]). Indeed, the optimal solution has always the pressure at the upper bound, since the cost

of the compressor is quite limited: according to the formulas used in [146] a compressor going from 35 to 50 bar only costs $0.074\$/Km^3$. This justifies the very low cost of the optimal configuration but at the same time shows the need of a more realistic objective function, further motivating our choice of a different objective function in the next subsection.

We stress that in Section 3.4 of [146] some experiments with different bounds on the feed side pressure and a different fresh feed pressure are reported. For the considered objective function, the different fresh feed pressure only implies a constant value added to the objective function due to the compressor on the inlet feed (from 1.05 to 35 this correspond to $0.736\$/Km^3$). Indeed, even adding this term our results outperform the results presented as the best in [146]. It is worth to notice that the best solution reported in [146] for this case is a 3-stage system, and not a 2-stage one, as in our case.

Table 3.11: Annual process cost expressed as USD/m^3. Case with free uniform pressures and free areas. Reference values recalculated from [146]. Case with fixed pressures is presented again for comparison.

	Reference	This work	
		Fixed pressures	Free pressures
GO	11.236 3 stages	9.115 3 stages	7.271 2 stages
ASPEN	11.660	9.095	7.262

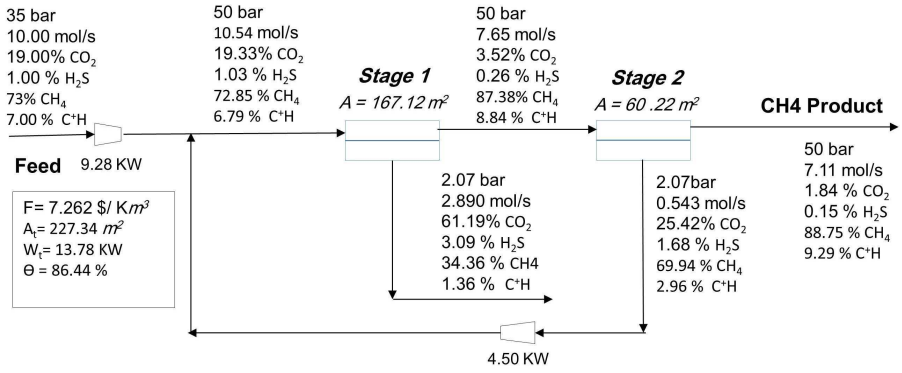


Figure 3.8: Optimal process for the case with free pressures and free area.

We feel the need to stress that the aim of the presented experiments was mainly focused at validating, first of all, our mathematical programming model, and just in a second step, our global optimization procedure in terms of quality of putative optima.

We are aware that the results used for comparison are not enough to guarantee that another global optimization procedure could not outperform our strategy. Nevertheless, as explained before, there is not paper in the current literature allowing such a comparison, for the missing of details on the global optimization procedure and/or missing parts in the description of the underlying mathematical programming models in terms of superstructure translation and objective function. We are looking forward to allow such comparison in our future research.

3.3.2 An industrial case-study

After validation, our optimization approach was applied to the recovery of CO_2 from blast furnace gas. Details on this part are more of interest for the membrane gas separation community, therefore here we will report just a very short summary (the interested reader can refer to [J1] for a detailed analysis) and some general considerations. With respect of the process parameters we considered:

- Feed = 1240 mol/s
- Gas pressure, P_{in} = 1bar
- Gas temperature, $T = 35°C$
- Gas composition and membrane permeabilities as in Table 3.12

Gas permeances are those reported for MTR's Polaris© membrane, used here as a reference of the best performance offered by polymeric membranes currently available in the market for CO_2 capture applications [125, 126].

The process product is the CO_2, that must be "cleaned" by impurities (N_2, in this case) to be re-used in industrial processes allowing the reduction of the carbon footprint of blast furnace gas (resulting from steel production). Product purity is expressed in terms of residual N_2 since this would be the only inert component when considering the transformation of the recovered CO_2 by catalytic processes [147]. The product is collected at the permeate side because its permeability is higher with respect to one of the other components (see second columns in Table 3.12).

The effect of CO_2 recovery and residual N_2 constraints on optimal process configuration, operation parameters (upstream and downstream pressures, and membrane surface) and separation cost is studied for CO_2 recoveries of 90, 95 and 99% while N_2 content is varied between 0.1, 0.5 and 1%.

Process configurations from 2 to 4 membrane stages were considered. The general superstructure (presented in Figure 3.3), is the same used for the validation with the reference case. It includes the possibility to have vacuum downstream pressures independent for each membrane stage by including a vacuum pump on each permeate stream. Upstream pressure is allowed to vary between 1 and 100 bar and this pressure is the same for all membrane stages while downstream pressure is allowed to vary between 0.2 bar and 1 bar and is independent for each stage.

Table 3.12: Feed composition and gas permeance

Gas	Composition (%mol)	Pure gas permeance (GPU)
$\mathbf{CO_2}$	23.2	1000
CO	22.6	20
$\mathbf{N_2}$	50.3	15
$\mathbf{H_2}$	3.9	85

The results of the optimization algorithm allowed to perform a full economical analysis on the process under consideration. For the sake of illustration, we present in Figure 3.9, all the putative optima for the case with three membranes and in Figure 3.10 the overall putative optima considering up to four membranes. Two points are worth to mention: first, the global procedure proposed some configurations presenting self-loops (some of the retentated flow of a membrane goes back in input to itself), such kind of configurations were not considered in the current state of the art as interesting, thus showing that removing from the superstructure the possibility of having self-loops removes some putative optima; second, the proposed solutions improved at least of 20% upon the results with respect to the simulation-based approach with the same or a shorter work-time (to give a general idea, the overall analysis with the global optimization procedure was performed in some weeks on a personal computer).

Furthermore, the method allowed us to perform a study on the effect of a possible decrease of membrane permeance or membrane selectivity on the optimal separation cost and process configurations, giving the possibility to show the possible impact of new materials on the overall process.

3.4 Conclusions

In this chapter, the membrane gas separation problem is introduced and a mathematical programming model proposed to allow the design of optimal configurations in terms of overall cost of the system. Then, our global optimization strategy is described in its main components and computational results are reported, showing that our method allows to improve upon a state of the art paper and it is able to provide results on a real case study (the CO_2 capture in steel production).

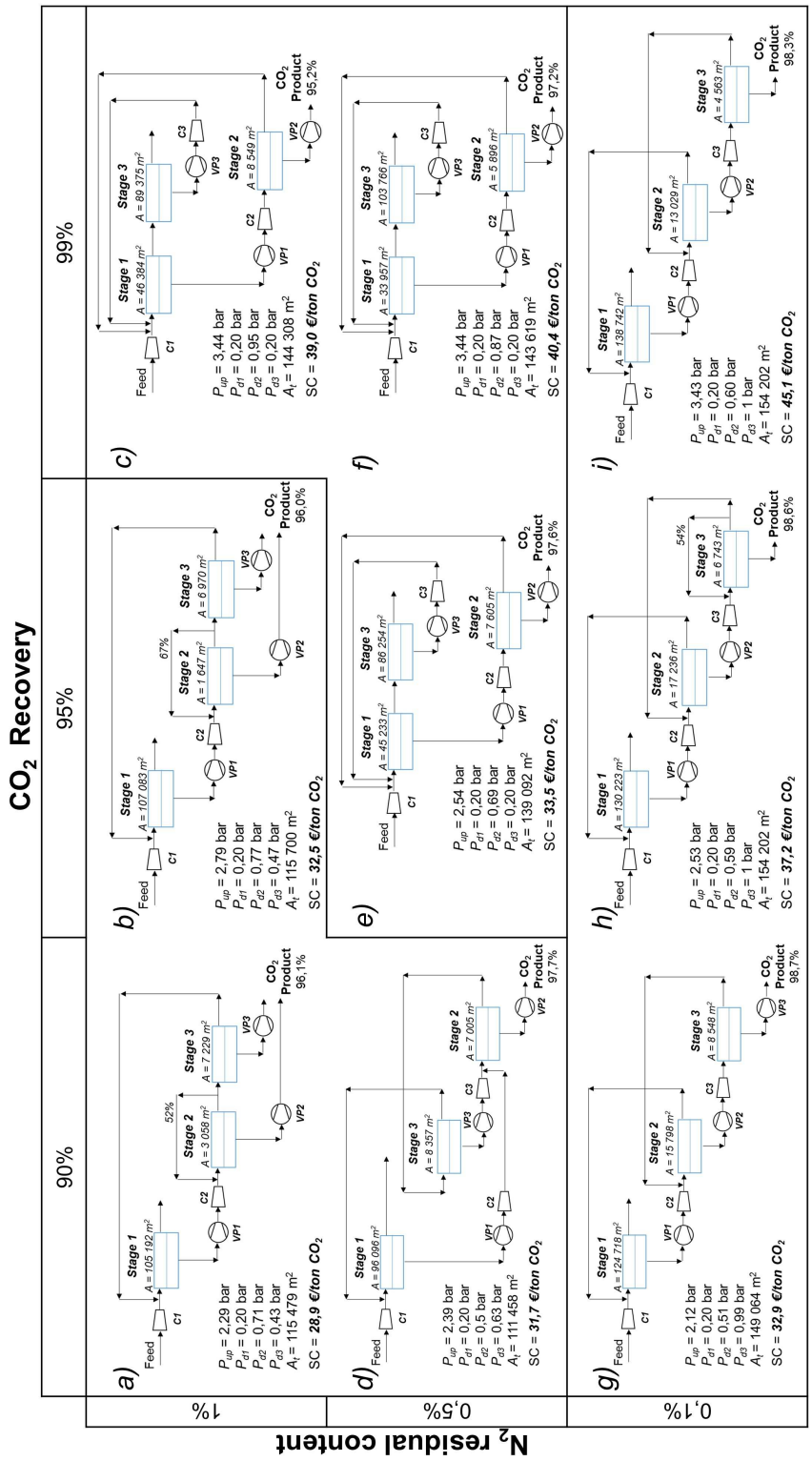


Figure 3.9: Best configurations when maximum number of stages is set to three. CO_2 recovery: 90, 95 and 99%, N_2 content: 0.1, 0.5 and 1%.

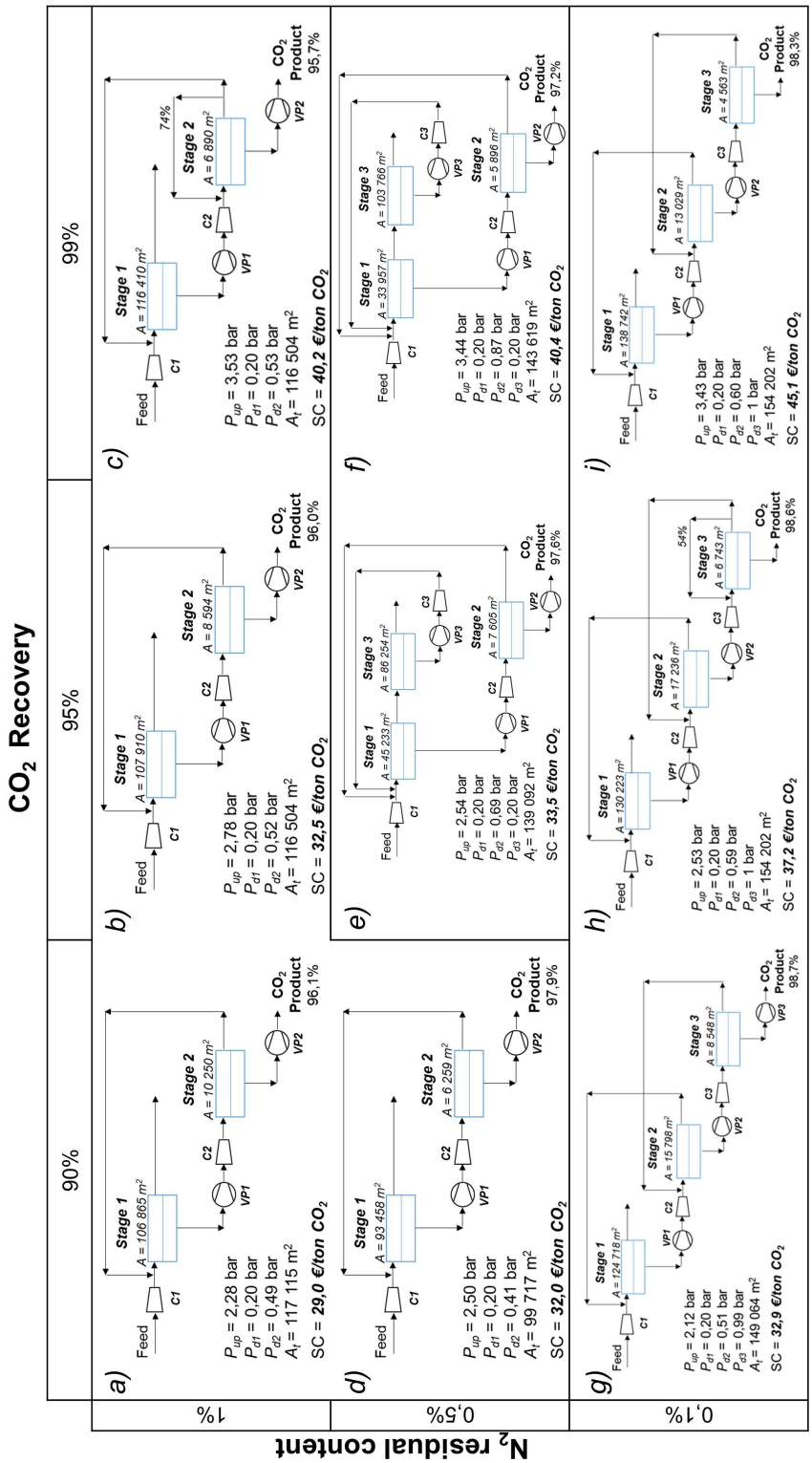


Figure 3.10: Overall best configurations when among processes with up to four membrane stages. CO_2 recovery: 90, 95 and 99%, N_2 content: 0.1, 0.5 and 1%.

4 Energy-aware network management

In the last years, energy consumption has been a central issue in many research fields. Energy is a central resource for the majority of industries and human activities (from heavy industry to the apparent "virtual" services supported by ICT, that consume a significant amount of the total worldwide energy consumption). The attention to energy consumption is due to two main factors: it is a limited and expensive resource and it has a strong environmental impact mainly due to the CO_2 emission associated with its production.

ICT plays a fundamental role in reducing the energy consumption of several human activities, yet, it is itself responsible for a significant portion of the global energy consumption. Although, year by year, ICT technologies are becoming more power efficient, that will not be enough to reverse the trend, making the ICT sector responsible for up to 23% of global green house gas (GHG) emissions in 2030 [12]. Therefore one of the main focal points for ICT resource management is energy-awareness, and optimization methods are a key tool to achieve it. Energy-aware management strategies minimize energy consumption by means of dynamical switching-on/off appliances (such as routers, computing resources) and, consequently, dynamically assigning service demands (traffic routing, assigning processes to servers, etc) to the current system configuration. To guarantee the viability of the found solutions, one fundamental issue is to have a suitable Quality of Services (QoS) level.

Among ICT application, telecommunication has a large impact on the world wide energy consumption: according to [103], Deutsche Telekom networks were responsible for around 0.5% of German yearly electricity expenditure, while the energy consumed by the largest providers such as AT&T or China Mobile reached 11 TWh per year in 2010 ([29]). Furthermore, projections for the consumption of medium sized operators like Telecom Italia and GRNET were around 400 GWh in 2015 ([29]). Quite recently, a global initiative headed by the GreenTouch consortium [85] to improve energy efficiency of telecommunications network, published a final report presenting all the main achievements [74] showing that energy efficiency in core network would improve by a factor 316 by 2020 by implementing and combining together the initiative findings.

IP networks can be made greener by working at different levels [124, 130]. Here the focus will be on the one where optimization can be beneficial, namely coordinating the management of the whole network infrastructure to optimize both energy consumption and performance of both routing and device configurations [178],[52],[46].

In the majority of the works, maximizing the *energy-efficiency* is equivalent to minimize the energy consumption. Practically speaking, the goal of the *Energy-Aware Network Management* (EANM) is to adapt network consumption to traffic levels. According to each specific proposal, this goal is achieved differently, e.g., by reducing the number

of active devices (line cards or routers), by smartly routing traffic, or by a combination of the two strategies. A challenging aspect of EANM is that despite the increase in energy efficiency, it should not degrade network performance or affect any Service Level Agreements existing in the network.

Our contribution to the EANM problem is of two natures:

- we introduce detailed optimization models to describe a large family of EANM problems. Using these tools, we proved the effectiveness of re-routing and dynamic switching on/off devices to obtain energy reduction. Furthermore, as we considered also protection strategies, we performed a detailed analysis of the trade-offs between energy consumption and resiliency. These tools can be used to effectively manage wide area networks;

- we performed an extensive analysis of the state of the art on EANM from an optimization point of view, putting in perspective the existing works and showing that a large part of them, even addressing a new application problem (namely, green networking) use indeed just well known optimization models. Furthermore, we showed that considering multi-periodicity is a key element in EANM problems;

- we proposed and analyzed the addition of valid inequalities and a new projected formulation for a given protection mechanism (namely the smart shared protection). This choice was motivated by the fact that the smart shared protection resulted the most energy-efficient, but also the most challenging to solve for state of the art solvers.

With the aim of detailing our contributions, the rest of this chapter is organized as follows: after a short introduction on some technical aspects related to EANM, the mathematical modeling description of the problem is described and a reduced version of our survey literature is reported with the aim of positioning our work with respect to the state of the art and to highlight its novelty, finally the main results we obtained from an energy reduction point of view are briefly discussed.

4.1 A literature review on Energy-aware Network Management

A very large body of work has been published on green networking, including a few survey papers. Even though there is a vast and specific literature on IP energy-aware management, and optimization plays a central role on it, our survey ([J7]) was the first to give a thorough review and analysis of all the different modeling features of the related optimization problems. The way EANM is performed relates to several elements: in the following, we report the ones that are necessary to describe our contribution and its position with respect to the state of the art.

Power profiles The power profile of a network device is the curve of power consumption versus traffic load. According to recent studies ([44, 119, 124, 161]), the power profile of current network devices and their main components, e.g. router chassis

and line cards, is almost independent of traffic load and not negligible energy savings can be achieved only switching off the devices (sleep mode). Therefore, power profiles are often approximated with an step-wise (ON-OFF) curve. It is worth to mention that some additional strategies based on adaptive link rate exist, similarly to CPU frequency scaling in data-centers ([54],[95]).

Routing and transport protocols Routing protocols have a major impact on traffic engineering techniques, and therefore on the energy management polices that can select traffic routes to reduce the consumption or to put to sleep some nodes or links (line cards).

In IP networks, we identify two main classes of routing protocols, i.e., flow based and shortest path based. Our contributions are on flow based protocols, in particular we based our model on the Multi-Protocol Label Switching (MPLS) protocol: each traffic demand (identified by a source and destination pair) is routed along one or multiple dedicated paths [172]. In this case, traffic engineering is very flexible because each routing path can be selected independently from the others by the network administrator.

Performance constraints Reducing energy consumption cannot be made at the cost of reducing the network performance. A common practice of network operators to ensure certain levels of QoS is to impose a maximum utilization threshold on both network routers and links, thus insuring that there is spare capacity to cope with unpredictable traffic variations and/or device failures. More elaborated strategies exist based on modelling congestion with increasing convex non-linear function (see, e.g., [73]) that however we did not consider in our work.

Network survivability and robustness Energy reduction and survivability/robustness are conflicting goals. Perfectly tailoring the active capacity to the incoming traffic level is clearly the natural way to maximally reduce network consumption. However, even though it is expensive from the energy perspective, enough spare capacity should always be available to cope with unexpected events, including device failures and unexpected traffic variations.

4.1.1 Energy-aware network optimization modeling

Energy Aware Network Management (EANM) problems are strictly related to another class of optimization problems called *Network Design* (ND). ND shares many modeling features with EANM, and it has been extensively studied in the last fifty years (see, for instance, [143], [16], and Chapter "An annotated bibliography in communication network design and routing" of [175]).

The presentation of the mathematical models is “modular” and “incremental”, starting from the basic EANM problem and then including one by one other network and problem features. This choice is motivated by two main reasons. First, in the literature different problems were addressed and different mathematical models were used: in this way it is easier to point out common and differences between existing works. Second, this allows

to highlight which features are common to ND problems, and therefore do not represent a new optimization problem, but just a new use of existing ones, and which are specific of EANM problems.

Basic energy-aware network management problem

Let us consider an IP network topology composed of routers and links connecting them. Each router consists of a chassis and a certain number of cards, such cards allow the connection, through the use of links, to other routers. The network can be represented by a directed graph $G = (V, A)$, where the set of nodes V represents the routers and the set of arcs A represents the links (and the cards). Let $c_{ij} \geq 0$ be the capacity associated with arc (link) $(i, j) \in A$. Traffic is represented by a set of demands D. Each traffic demand $d \in D$ is described by a source node o^d, a destination node t^d and a non-negative bandwidth request r^d.

The *IP-Basic Energy-Aware Network Management* (IP-BEANM) problem asks to route the set of demands so as to minimize the network energy consumption. To represent the flow of demand d on each arc (i, j), we introduce non-negative variables f_{ij}^d.

To avoid the use of too many variable/parameters names, with some abuse of notation, we use the same name (but different set of indeces) for entities that have a strong connection. For example, to allow a compact representation for different, strictly related, model variants we introduced the additional flow variables f_{ij} and f_i that represent the total traffic carried by link $(i, j) \in A$ and the total traffic entering node $i \in V$, respectively.

Different functions have been used to model network energy consumption. Energy consumption is commonly represented as the sum of the link energy consumption $\Pi_{ij}(f_{ij})$ and the node energy consumption $\Pi_i(f_i)$. In IP-BEANM such functions are assumed to be continuous and non-decreasing. The general notation is summarized in Table 4.1.

The model for IP-BEANM is the following:

Table 4.1: Elements of the basic problem

Sets	
V	Network nodes
$A \subset V \times V$	Network links
D	Traffic demands
Parameters	
o^d	Origin of demand d
t^d	Destination of demand d
r^d	Traffic request for demand d
c_{ij}	Capacity on link (i,j)
$\Pi_{ij}(f_{ij})$	Consumption function for link (i,j)
$\Pi_i(f_i)$	Consumption function for node i
Variables	
f_{ij}^d	Amount of flow of demand d and on link (i,j)
f_{ij}	Total amount of flow carried by link (i,j)
f_i	Total amount of flow carried by node i

$$\min \sum_{(i,j)\in A} \Pi_{ij}(f_{ij}) + \sum_{i\in V} \Pi_i(f_i) \tag{4.1}$$

$$\sum_{\substack{j\in V:\\(i,j)\in A}} f_{ij}^d - \sum_{\substack{j\in V:\\(j,i)\in A}} f_{ji}^d = \begin{cases} r^d & \text{if } i = o^d \\ -r^d & \text{if } i = t^d \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in V,\, d \in D \tag{4.2}$$

$$\sum_{d\in D} f_{ij}^d = f_{ij} \qquad \forall (i,j) \in A \tag{4.3}$$

$$\sum_{(j,i)\in A} f_{ji} + \sum_{\substack{d\in D:\\o^d=i}} r^d = f_i \qquad \forall i \in V \tag{4.4}$$

$$f_{ij} \le c_{ij} \qquad \forall (i,j) \in A \tag{4.5}$$

$$f_{ij}^d \ge 0 \qquad \forall (i,j) \in A,\, d \in D \tag{4.6}$$

$$f_{ij} \ge 0 \qquad \forall (i,j) \in A \tag{4.7}$$

$$f_i \ge 0 \qquad \forall i \in V. \tag{4.8}$$

Objective function (4.1) minimizes the network power consumption due to the flow on arcs and through nodes. Flow conservation constraints (4.2) ensure that each traffic demand is routed from its source to its destination, while constraints (4.3) and (4.4) compute the bandwidth requirements for links and nodes, respectively. Link capac-

ity are enforced by constraints (4.5). Inequalities (4.6)-(4.8) define the domain of the variables. Different formulations can be taken into account without significantly changing the structure of the problem (for example, by considering the in-out flow for node capacity and not only the in-flow).

If only arc energy consumption is considered and functions $\Pi_{ij}(\cdot)$ are linear, the model reduces to a *multicommodity flow problem* (see Chapter 17 of [5]). We recall that the multicommodity flow problem with continuous flow variables can be polynomially solved. Instead, the *multicommodity integral flow problem* (where flow variables are integer) is NP-complete in its decision form, even if only two commodities and unitary capacities are considered (see Appendix of [80]). We observe that, in the multicommodity model, cost and capacity are associated with arcs. However, also a problem in which cost and/or capacity are defined for nodes can be modeled as a multicommodity flow problem by modifying the graph as follows. Each node i is replaced by two nodes $i^{'}$, and $i^{''}$, one representing the input side of i, the other representing the output side. An arc between $i^{'}$ and $i^{''}$ is added and the capacity and the cost of node i are associated with it. The size of the problem will grow to $2N$ nodes and $A + N$ arcs (see Chapter 2 of [5]).

Besides linear objective functions, non-linear, both convex and non-convex functions [81, 162, 38], are considered in the literature. Piece-wise linear approximations may be applied when convex functions are minimized, (see e.g., [81, 73]), still preserving linearity of the formulation. Convex non-linear functions have also been used to model other kind of costs/objectives such as delays in the urban networks or congestion in communication networks and such applications are widely addressed in the literature (see Chapter 14 of [5] and [70]).

Single-path routing

To simplify network management tasks and avoid complex operations such as packet reordering, network administrator prefer to adopt a single path routing scheme. Binary variables $x_{ij}^d \in \{0, 1\}$ are introduced, which are equal to one if link (i, j) is used by demand d and constraints (4.2)-(4.3) must be modified as follows:

$$\sum_{\substack{j \in V: \\ (i,j) \in A}} x_{ij}^d - \sum_{\substack{j \in V: \\ (j,i) \in A}} x_{ji}^d = \begin{cases} 1 & \text{if } i = o_d, \\ -1 & \text{if } i = t_d, \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V,\, d \in D \tag{4.9}$$

$$\sum_{d \in D} r^d x_{ij}^d = f_{ij}, \quad \forall (i, j) \in A \tag{4.10}$$

Equations (4.10) determine the total flow carried by each arc[1].

[1]Usually, the relation expressed by constraints (4.10) is used to remove f variables, but here we prefer to keep both variables to avoid modifying the other constraints involving variables f

Maximum utilization constraint

Network congestion is usually prevented by the network operator by means of a maximum link utilization threshold $\mu_{ij} \leq 1$. In this case capacity constraints (4.5) become:

$$f_{ij} \leq \mu_{ij} c_{ij} \quad \forall (i,j) \in A \tag{4.11}$$

This modification does not change the structure of the problem, as it is simply a scaling factor of the arc capacity. From now on, we will consider always the constraint with congestion reduction mechanism [2].

Sleeping capabilities

As mentioned before, the consumption profile of current network devices is quite well approximated by an ON-OFF (step) curve. Therefore, most papers ignore utilization-based energy costs and considers only a fixed consumption π_i for each active node $i \in V$ and π_{ij} for each active link $(i,j) \in A$. Binary variables y_i and w_{ij} are introduced which are equal to 1 if node $i \in V$ and link $(i,j) \in A$, respectively, are powered on. The objective function can be expressed as:

$$\sum_{(i,j) \in A} \pi_{ij} w_{ij} + \sum_{i \in V} \pi_i y_i \tag{4.12}$$

Capacity constraints (4.5) must be modified and additional constraints must be included:

$$f_{ij} \leq \mu_{ij} c_{ij} w_{ij} \quad \forall (i,j) \in A, \tag{4.13}$$

$$w_{ij} \leq y_i, \quad \forall (i,j) \in A \tag{4.14}$$

$$w_{ij} \leq y_j, \quad \forall (i,j) \in A \tag{4.15}$$

Constraints (4.13) prevent routing a demand through a sleeping link, and replace constraints (4.5). Constraints (4.14)-(4.15) forbid the activation of a link connected to a sleeping router. If $\pi_i = 0$, the second term of the objective function can be neglected and all routers are assumed to be always active: constraints (4.14) and (4.15) can be neglected, as well. In this case, the resulting model shares the structure with the Network Design problem, where a minimum cost subset of arcs must be selected (see [143]).

Bundled links

Each link is normally made by a given number of cards with the same capacity and the same power consumption. Each line card can be switched on or off independently from the others. To model this feature, binary variables w_{ij} can be substituted with integer variables: $w_{ij} \in [0, \ldots, n_{ij}]$, representing the number of active line cards on each link. Constraints (4.14) and (4.15) should be slightly modified accordingly:

[2] The case with no congestion reduction is equivalent to set $\mu_{ij} = 1$

$$w_{ij} \leq n_{ij} y_i, \quad \forall (i,j) \in A \tag{4.16}$$
$$w_{ij} \leq n_{ij} y_j, \quad \forall (i,j) \in A \tag{4.17}$$

The Network Loading problem with modular capacity, namely the problem of allocating the minimum cost number of modular capacity devices on network arcs, has a similar model structure (see [143]).

Protection from single link failures

The need for greener networks should not compromise the network capability to react to unexpected events such as device failures. Failure events are usually handled by automatic protection mechanisms that redirect the traffic over still working routes.

From a failure dependency perspective, survivability policies can be classified into *restoration* and *protection* techniques. In restoration policies, the alternative routing depends on the fault, while in protection policies re-routings are fault-independent and, for each demand, an alternative disjoint path is available (the so called backup path).

Restoration policies may provide more flexibility, but they may require to re-route all the demands, including the ones not affected by the current failure. Moreover, being fault-dependent, they may require additional information other than the affected demands, e.g., the failed elements. The most common restoration/protection schemes are *global rerouting* (or unrestricted restoration), *line restoration* (or local rerouting) and *path restoration* (or end-to-end restoration). They differ in the number of demands to reroute (all or just the ones affected by the fault) and in the information needed (affected demands or affected demands plus failed components).

From the capacity usage point of view, survivability policies are classified into *dedicated* and *shared mechanisms*. In a dedicated protection scheme, a given amount of capacity is reserved to reroute each demand. On the contrary, in a shared protection mechanism the backup capacity can be shared by demands not affected by the same failures. This allows to reduce the needed backup capacity but leads to more computationally challenging design problems. The choice of the restoration/protection scheme is also influenced by the routing protocol.

Most of the protection/restoration schemes have been widely addressed in the optimization literature. The complexity of the pricing problem for the path formulation of survivable network design with several protection/restoration mechanisms is investigated in [140]. For an overview of survivable networks and related mechanisms we address the reader to [24, 99, 100].

Single link failure point-to-point protection In our work, we focused on path protection (point-to-point) schemes. In the following we show how dedicated and shared protection can be modelled. Furthermore, using some energy considerations, we introduce a variant of these approaches that has impact on the overall energy-consumption. To model point-to-point protection, backup paths variables and flow conservation constraints are needed. Let ξ_{ij}^d be the binary variables which are

equal to 1 if link $(i,j) \in A$ is used by the backup path of demand $d \in D$. The additional constraints to represent the protection scheme are:

$$\sum_{\substack{j \in V: \\ (i,j) \in A}} \xi_{ij}^d - \sum_{\substack{j \in V: \\ (j,i) \in A}} \xi_{ji}^d = \begin{cases} 1 & \text{if } i = o_d, \\ -1 & \text{if } i = t_d, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall i \in V,\, d \in D \tag{4.18}$$

$$x_{ij}^d + \xi_{ij}^d \leq 1, \quad \forall (i,j) \in A,\, d \in D \tag{4.19}$$

$$x_{ij}^d + \xi_{ji}^d \leq 1, \quad \forall (i,j) \in A,\, d \in D. \tag{4.20}$$

Constraints (4.18) are the flow conservation for the backup paths, while (4.19)-(4.20) guarantee that primary and backup paths are link disjoint. It is necessary to modify capacity constraints to take into account backup flows. Each end-to-end protection scheme is characterized by a different way to compute the total link flow.

Dedicated protection The *dedicated* protection scheme reserves the same amount of bandwidth on each link of the primary and the backup paths, then the dimensioning constraint (4.10) is replace by:

$$\sum_{d \in D} r^d \left(x_{ij}^d + \xi_{ij}^d \right) = f_{ij} \quad \forall (i,j) \in A \tag{4.21}$$

Shared protection The *shared* protection aims at reducing the reserved backup capacity by allowing backup paths to share resources. As in protection mechanisms, a single backup path is reserved, the backup capacity for each link is forced by the failure scenario which requires the highest possible backup capacity, i.e. the failure that affects the highest amount of demands.

Binary variables g_{ijkl}^d are introduced, which are equal to 1 if traffic demand $d \in D$ is rerouted on link $(i,j) \in A$ when a failure occurs on link $(k,l) \in A$ (i.e. if traffic demand $d \in D$ uses a primary and a backup paths routed, respectively, on link $(i,j) \in A$ and link $(k,l) \in A$). The dimensioning constraint (4.10) is replace by:

$$g_{klij}^d \geq x_{kl}^d + \xi_{ij}^d - 1, \quad \forall (i,j),(k,l) \in A,\, d \in D, \tag{4.22}$$

$$\sum_{d \in D} r^d \left(x_{ij}^d + g_{klij}^d \right) = f_{ij} \qquad \forall (i,j),(k,l) \in A. \tag{4.23}$$

Constraints (4.22) force the correct values for variables g_{ijkl}^d, i.e. equal to 1 when the primary path uses link (k,l) and the backup path uses link (i,j), while constraints (4.23) calculates the overall link dimensioning due to primary and backup paths. We can observe that this last one is a bottleneck constraint, as the value of f_{ij} depends on the worst case scenario among all link failures ($\forall (k,l)$).

The smart consumption variant for protection The smart protection variant exploits the possibility of reactivating sleeping line cards in a few milliseconds ([91]) and therefore the possibility of putting to sleep line cards that carry only backup paths during normal network operation. It is then assumed that line cards used only by backup paths are powered only when required by the occurrence of a failure, thus having negligible consumption.

Such new feature of dynamic network can be modeled by restoring constraints (4.10) at the place of constraints (4.21) for the dedicated protection (and constraints (4.23) for the shared one). To guarantee that the total available capacity on each link (all cards n_{ij} are switched on) is not exceeded by the sum of primary and backup traffic routed on it, for the dedicated protection the following constraints must be added:

$$\sum_{d \in D} q_d \left(x_{ij}^d + \xi_{ij}^d \right) \leq \bar{\mu}_{ij} n_{ij} y_j, \quad \forall (i,j) \in A. \tag{4.24}$$

Similarly, for the shared protection case the following constraints must be added:

$$\sum_{d \in D} q_d \left(x_{ij}^d + g_{klij}^d \right) \leq \bar{\mu}_{ij} n_{ij} y_j, \quad \forall (i,j),(k,l) \in A \tag{4.25}$$

We can observe that the status of cards (and therefore their energy consumption) is forced by primary paths only, but even the sleeping cards carrying only backup paths have to be connected to an active chassis.

Since the occurrence of a link failure is very unlikely, we opted to include a second higher utilization threshold ($\bar{\mu} > \mu$) enabled only when backup resources are exploited. Allowing the network to operate with a higher but still reasonable congestion during the very limited failure intervals allow to further increase the energy savings.

It is worth pointing out that the same strategy (switching on when a failure occurs) cannot be applied to network routers because a chassis switch on requires a significant time.

Robustness Another important aspect concerns the intrinsic uncertainty which affects traffic demands (see for example [C4],[56]). In our work, we presented a robust version of our models using the cardinality-based approach proposed in [23], nevertheless, to keep the presentation compact, we decided to avoid a detailed description on this aspect[3]. The interested reader can refer to our literature review [J7] for robustness on IP green networks, and to [22] and [21] for general references on robustness.

We illustrate with an example the impact of the considered protection features (see Figure 4.1)

[3]As we reported the original pictures taken from our works, in some of them, some results on the robust case will appear anyway.

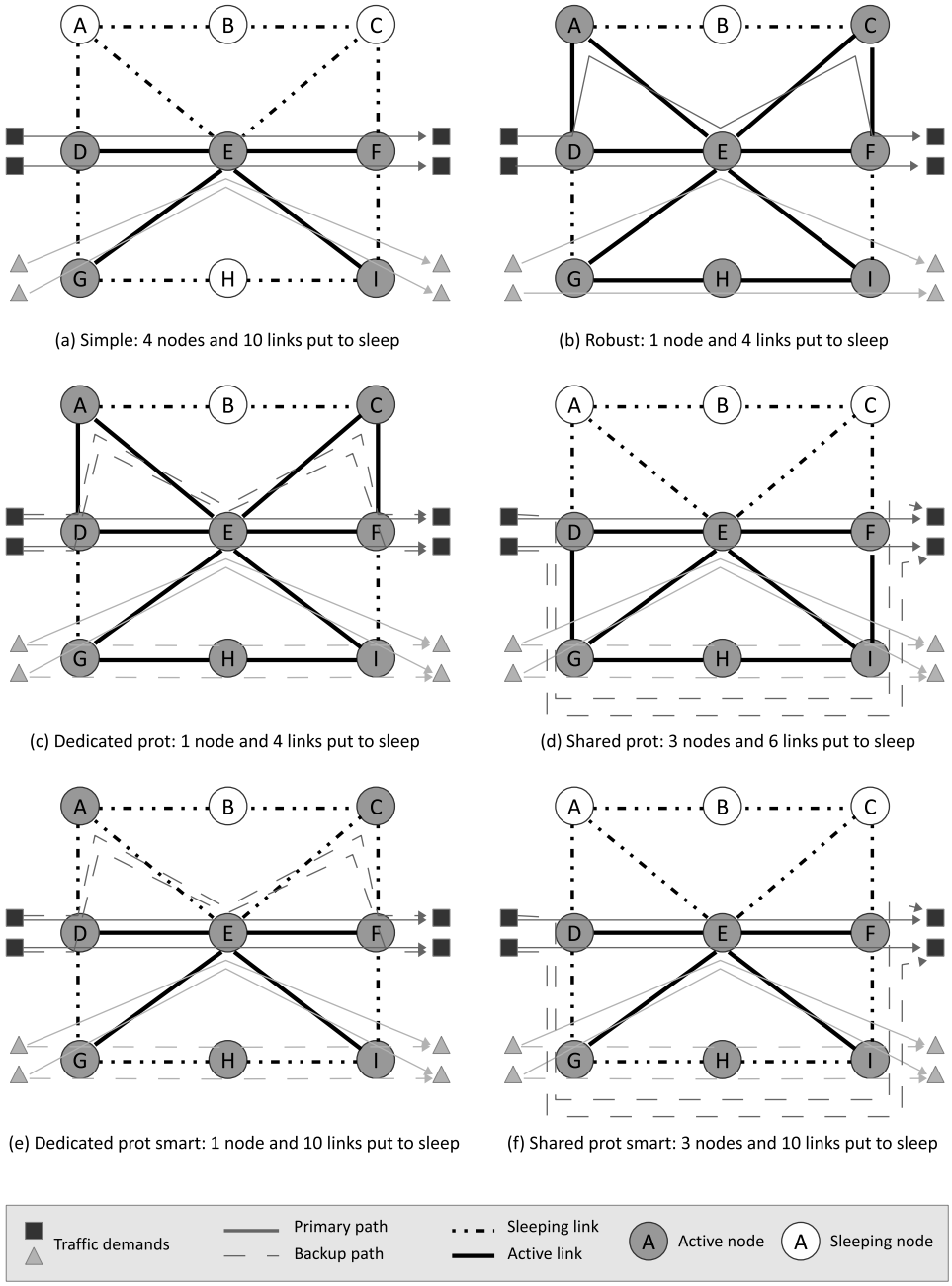


Figure 4.1: Energy consumption minimization vs Resilience requirements.

In the figures, link capacity is assumed to be two units and there are four traffic demands each requesting one unit of traffic.

Figure 4.1(a) represents the simple case for which no protection schemes are implemented. We can see that there are 4 nodes and 10 bidirectional links to put to sleep, making this case the most energy-efficient.

In case of dedicated protection (Figure 4.1(c)), additional links and nodes have to be switched on (three more nodes and six more links) to carry the backup paths. Note that, since each backup path has the same bandwidth requirement of the primary one, the two demands $G - I$ cannot be routed on the links already used by the backup paths of the two demands $D - F$. However, by implementing shared protection (Figure 4.1(d)), it is possible to sensibly reduce the consumption due to protection and put to sleep two more nodes and two more links w.r.t. the dedicated case. These savings can be achieved because shared protection allows to share the backup resources on the links used by the backup paths (links $G - H$ and $H - I$), since the primary paths of demands $D - F$ are link disjoint from the primary paths of demands $G - I$.

As shown in Figures 4.1(e) and 4.1(f), the smart protection allows to further reduce the network consumption by switching off an additional number of links. With respect to the corresponding classic cases, with the dedicated-smart protection six more links can be put to sleep, whereas with shared-smart protection the additional sleeping links are four. With the smart strategies, it can be advantageous to use different links to carry primary and backup paths, as this allows to switch off link devices that are used only for backup.

Multi-period network management

In IP networks, traffic conditions are not constant but vary throughout the day. Accordingly, the network configuration can be adjusted to accommodate the incoming level of traffic. Each change of configuration may result in an additional energy consumption due to reactivation of sleeping elements. Furthermore, the number of changes can be limited for instance by constraining the changes of routing paths or the number of reactivation for a specific device: a too frequent state switching could negatively affect the device lifetime. The resulting multiperiod problem consists in planning network configuration along a given time horizon, so as to minimize the overall network energy consumption. As shown in some recent studies [47, 48], due to the slow dynamic of Internet traffic, a few time periods with a duration in the order of hours are enough to provide a good representation of traffic.

To extend the models to take into account multiperiodicity, let S denote the set of time periods in the considered time horizon and let $r^{d\sigma}$ denote the amount of traffic of demand d in time period σ.

If the routing can change in each time period, single period path variables x_{ij}^{d} must replaced by variables $x_{ij}^{d\sigma}$, which are equal to 1 if link $(i, j) \in A$ is used to route demand $d \in D$ during period $\sigma \in S$. Routing constraints should be defined for each single time period:

$$\sum_{\substack{j\in V:\\(i,j)\in A}} x_{ij}^{d\sigma} - \sum_{\substack{j\in V:\\(j,i)\in A}} x_{ji}^{d\sigma} = \begin{cases} 1 & \text{if } i = o^d \\ -1 & \text{if } i = t^d \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V,\, d \in D,\, s \in S \tag{4.26}$$

$$\sum_{d\in D} r^{d\sigma} x_{ij}^{d\sigma} = f_{ij}^{\sigma} \quad \forall (i,j) \in A,\, \forall s \in S \tag{4.27}$$

$$\tag{4.28}$$

All the other variables are changed accordingly. Inter-period constraints have to be considered, which account for the energy consumption due to device reactivation, limit in the number of reactivations, or in the number of changes in routing paths.

Let non-negative real variables z_j^σ denote the energy consumed by chassis $j \in V$ if it is activated at the beginning of period $\sigma \in S$ and binary variables u_{ijk}^σ denote the activation state of line card k: they are equal to 1 if the k-th line card of link $(i,j) \in A$ is activated from the sleeping state at the beginning of period $\sigma \in S$.

The value of variables z is computed accordingly to the following constraints:

$$z_j^\sigma \geq \delta \pi_j \left(y_j^\sigma - y_j^{\sigma-1} \right), \quad \forall j \in V,\, \sigma \in S, \tag{4.29}$$

where δ is a parameter in $[0,1]$ representing the additional fraction of the nominal chassis consumption which is consumed to reactivate chassis $j \in V$.

Being η_{on} the maximum number of reactivations allowed for each line card along the whole time-horizon, the switching limit is imposed by means of:

$$\sum_{k=1}^{n_{ij}} u_{ijk}^\sigma \geq w_{ij}^\sigma - w_{ij}^{\sigma-1}, \qquad \forall (i,j) \in A,\, \sigma \in S \tag{4.30}$$

$$\sum_{\sigma\in S} u_{ijk}^\sigma \leq \eta_{on}, \quad \forall (i,j) \in A,\, k \in \{1,\ldots,n_{ij}\}. \tag{4.31}$$

Eq. (4.30) forces the model to reactivate the required number of line cards at the beginning of each time period, while Eq. (4.31) impose the reactivation limit for each single line card.

Additional constraints limiting the number of path adjustment can be present.

4.1.2 Our contribution to the EANM

Some of the most popular studies on EANM, [49, 51, 50, 52], explicitly address the variant of the IP-BEANM problem (4.1-4.8) adapted to model sleeping capabilities. Therefore, in these papers routing is splittable. Furthermore, a single line card is considered. Note that in the proposed MILP instead of the node activation constraints (4.14-4.15), a

big-M formulation is used[4]:

$$\sum_{(i,j)\in A} w_{ij} + \sum_{(j,i)\in A} w_{ji} \leq My_i, \qquad \forall i \in V. \tag{4.32}$$

The MILP formulation presented in [49, 51, 50] is solved only in [52], where the authors propose some little refinements to reduce the MILP complexity, such as switching to a *per-source routing scheme*. To the best of our knowledge, we were the first to consider a multiperiod model. The only exception is [76] (and its extention -[75]), where nevertheless, only two macro-periods were considered (low and high demands), no bundle link are take into account and only the results of a heuristic approach are reported. In the context of energy-aware network management, taking into account explicitly in the model the periodic nature of the demands and therefore optimizing along a full planning horizon is of paramount importance. From an application point of view, as the switching on/off of routers is associated to energy consumption, considering a multiperiod model allow to take into account also their contribution to the overall energy consumption, and therefore to obtain solutions that are - in general - better than the ones obtained optimizing period by period. Furthermore, multiperiod models allow to impose limits on the number of device transitions. As we already mentioned, this is important, as frequent changes in status of the devices reduces their lifetime, and therefore must be limited. From an optimization point of view, only the addition of multiperiodicity produces models that are different from the classical network design and network loading models, and therefore produces a novelty from the optimization point of view in terms of mathematical models.

In [J11], we introduced two MILP exact formulations, the first assumes a fixed routing configuration along the whole set of time periods, and the second allows the network administrator to modify the routing path at the beginning of each time interval. Multiple line cards are considered on each link (bundled links), maximum utilization constraints are imposed, energy savings are achieved by putting to sleep line cards and router chassis and by limiting the reactivation of the latter (reactivation causes a consumption spike). Furthermore, limitations on the maximum number of times that a single line card can be switched on within the whole set of periods are enforced to preserve the line card lifetime (see Section 4.1.1). In [C6] a novel heuristic to quickly find a sub-optimal solution for the two previous formulations is proposed. The algorithm is based on a GRASP methodology that sequentially solves a modified version of the basic formulation for a single traffic demand. Another heuristic is presented in [J11], where the configuration of each single time period is obtained by solving a single-period version of the reference MILP. While the exact MILP and the GRASP-based algorithm are intended for offline centralized approaches only, the single period heuristic can be applied in an online fashion, by solving the single-period problem at the beginning of each time interval or every time the network becomes too saturated or too empty. While there is a substantial literature on the integration, at the optical layer, of protection techniques with energy-aware network management practices, only a few studies have dealt with this problem at the IP layer [7, 118][C5][C7] and [J12] at the time of writing our survey.

[4]this formulation is dominated by the one that uses constraints (4.14)-(4.15).

In the preliminary work presented in [C5] and [C7], the multi-period problem formulations discussed in [C6] and [J12] are integrated to support, respectively, dedicated protection and shared protection. Both papers propose a MILP formulation and adapt the single-period heuristic presented in [J12] to the protected problems. We compare the *classic* approach where all network elements, carrying at least a primary or a backup path, are powered-on, and our proposed *smart* version (that allows the switching-off of line cards crossed by backup paths exclusively). The problem addressed in [C6] and [J11] is further extended in [C4] to account for robustness from traffic variations. The base formulation is modified to integrate the cardinality constrained method first proposed in [22] to manage traffic demands which are uncertain within a close symmetric interval. In [J12] a comprehensive formulation considering protection and robustness is introduced and it is exploited to conduct an overall trade-off evaluation between energy-efficiency and network survivability.

All the above formulations are suitable for a centralized offline approach, while the single-time period heuristic adapted to each specific problem variant can be employed in an online manner. Both our off-line and on-line strategies are centralized. This is motivated by the fact that we wanted not only to produce an algorithm to manage the network, but also to provide a tool to evaluate the maximal energy reduction that can be achieved using the on-off of devices and the re-routing of demands [5].

To put in perspective our contribution, and to summarize the results of the classification, in Table 4.2 is reported a reduced version of the summary of the literature presented in our survey work [J7]. Only the lines corresponding to our contribution are reported (as a consequence of this reduction, also some unnecessary columns were removed).

In short, at the time of the writing of the survey, excluding our work, only 9 papers present both the "Node off" and "Bundled link" features, but only 5 of them do not use the simplifying assumption of flow proportial consumption ([106, 131, 174, 118, 107]). As we explained before, current devices exhibits an on/off consumption profile, furthermore, link consumption depends on the number of active cards, therefore the "Bundled link" feature is necessary to take into account correctly the consumption. The single-path routing (necessary to have solutions that can be implemented with some protocols, for example the MPLS) is presented in other 12 works at exclusion of ours, but only 7 of them do not use the simplifying assumption of "Flow proportional consumption". Furthermore, if the "Node off" feature is asked, only 4 works remain ([17, 76, 10, 75]). There were no other works but ours that considered "Node-off", "bundled links", "no flow proportional consumption" and "single-path routing" features altogether. The other main contribution that distinguish our works from the others is to take into account multiperiod models, that as we explained before, is of central importance to model the problem correctly and allowing to obtain significant energy consumption reductions.

[5] A detailed analysis of on-line and off-line methods, and centralized vs decentralized methods, with a discussion on the advantages and disadvantages is presented in [J7].

Papers	Nodes off	Bundled Links	Multiperiod	Protection	Single-path routing	Robustness
[J11]	x	x	x		x	
[C6]	x	x	x		x	
[C5][C7]	x	x	x	x	x	
[C4]	x	x	x		x	x
[J12]	x	x	x	x	x	x

Table 4.2: Extraction of our contribution of the literature classification on flow-based routing

4.2 Experimentally comparing different resilience strategies

We present here some computational tests that allows to compare the different resiliency strategies in terms of energy consumption and network congestion.

First we introduce our test-bed and instance characteristics, then a description of the results on each strategy is discussed and, finally, extensive results on the largest instances are analyzed. For the shared protection mechanism, even apparently small instances are difficult to solve by state of the art solvers, in these cases heuristic strategies are applied to obtain good quality solutions. All the experiments were carried out on machines equipped with Intel i7 processors with 4 core and multi-thread 8x, and 8Gb of RAM.

The test-bed

We tested both exact and heuristic methods using four network topologies provided by the SND Library (SNDLib) [141], i.e. `polska`, `nobel-germany`, `nobel-eu` and `germany`. The network nodes are equally and randomly divided between core routers and edge routers. Notice that only core routers can be put to sleep, since they are neither source nor destination of any traffic demand. In Table 4.3, columns $|N|$, $|N_c|$, $|A|$ and $|D|$ represent the number of nodes and core nodes, the number of unidirectional links and the number of traffic demands, respectively.

Table 4.3: Test instances - Networks

Network	$\|N\|$	$\|N_c\|$	$\|A\|$	$\|D\|$
polska	12	6	36	15
nobel-ger	17	9	42	21
nobel-eu	28	14	82	90
germany	50	25	176	182

In each test instance all routers are assumed to be equipped with the same type of chassis and the same type of cards. However, we experimented with three different configuration cases, *alfa*, *delta*, and *eta*, wherein the chassis technology is always the

Table 4.4: Overview of different network configurations

case	device	capacity	hourly cons.
–	Chassis Juniper M10i	16Gbps	86.4 W
$alfa$	FE 4 ports	400 Mbps	6.8 W
$delta$	OC-3c 1 port	155 Mbps	18.6 W
eta	GE 1 port	1 Gbps	7.3 W

same, while the type of cards is varied (but the same technology is used for all the cards in a given instance). Chassis and card details are reported in Table 4.4.

We recall that traffic varies along the planning horizon. Each single day in split in six traffic periods corresponding to the following time intervals: 1) 8a.m.-11a.m., 2) 11a.m.-1p.m., 3) 1p.m.-2.30p.m., 4) 2.30p.m.-6.30p.m., 5) 6.30p.m.-10.30p.m., 6) 10.30p.m.-8a.m. In each interval a different profile is considered with respect to a nominal value. In Figure 4.2, the fractional value of the demand with respect to the nominal value is reported.

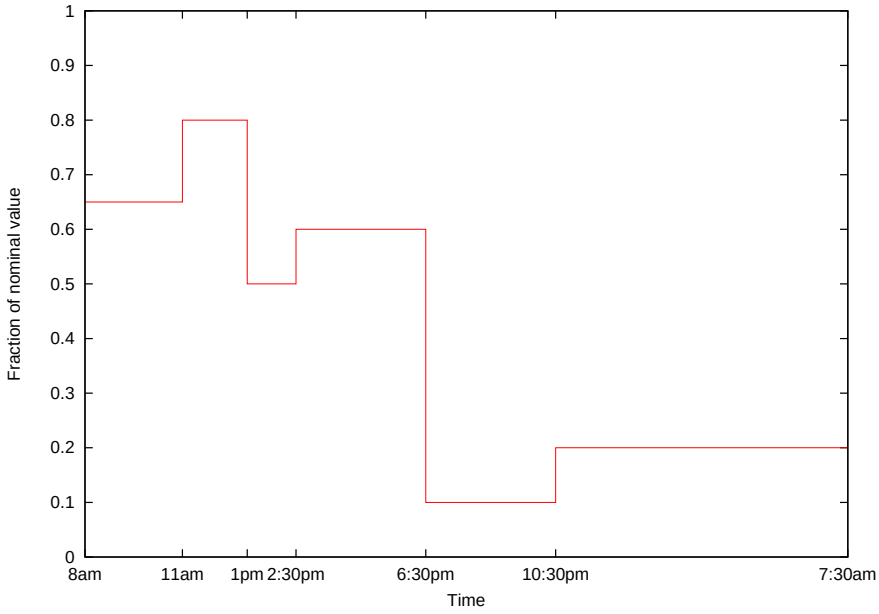


Figure 4.2: Traffic demands according to scenarios'profiles

To derive traffic matrices, we used as nominal values the ones provided by the SNDLib and we scale them with a fixed parameter $\varpi_{\mu}^{\bar{\mu}}$ to obtain a worst case scenario: a sort of "saturated" network in the peak hours. Indeed, the chosen value of $\varpi_{\mu}^{\bar{\mu}}$ is the highest value such that the matrix obtained multiplying the SNDLib values by it can be routed in the real full active network with protection (dedicated or shared), while respecting

the maximum utilization in normal conditions μ, and the maximum utilization in failure conditions $\bar{\mu}$. In the majority of our tests we used matrices scaled for $\varpi_{50\%}^{85\%}$ computed by considering dedicated protection. That is, we used μ (link max-utilization due to primary paths) equal to 50% and $\bar{\mu}$ (link max-utilization due to both primary and backup paths) equal to 85%.

We experimented with three different traffic scenarios. They were generated by considering traffic values (q_d^σ) as follows:

$$q_d^\sigma = r_d^\sigma \rho_d$$

where parameter r_d^σ is generated according to the uniform distribution:

$$\mathcal{N}(\overline{r}_d^\sigma + \hat{r_d^\sigma}, \overline{r}_d^\sigma - \hat{r_d^\sigma})$$

The average values $\overline{r}_d^\sigma$ were chosen according to the traffic profile of Figure 4.2, the variation $\hat{r_d^\sigma}$ is chosen as 0.2 (negative values are rounded up to value zero).

Each resulting instance is the combination of a network topology, an equipment configuration and a traffic scenario (for example: *p*olska with card of type *delta* and scenario 2).

As for the remaining parameters, we set δ (chassis switching-on normalized consumption) equal to 0.25, ε (switching-on limit) equal to 1, and n_{ij} (number of cards in link (i, j)) equal to 2 for each link.

Savings vs. Protection

First we aim at pointing out the impact of the different features provided according to the protection (and robustness) strategy considered, i.e. *simple*, *robust*, *dedicated-classic*, *shared-classic dedicated-smart*, *shared-smart*, *robust plus dedicated-classic* and *robust plus dedicated-smart*. The expected trade-off between energy savings and network survivability is reported in Figure 4.3.

Starting from the simple energy-aware problem with no protection and no robustness, we expect the energy consumption of the network to gradually increase if we demand an higher protection/robustness level. At the first level, we put the *robust* approach with no protection, which, by varying the robustness parameters, allows to allocate additional resources to cope with traffic variations. Then, we find, in sequence, the *shared-smart* strategy and the *dedicated-smart* one. Although shared protection guarantees the same degree of survivability of the dedicated one, with single link failures, we consider it less conservative because it produces solutions with, in general, less spare capacity available. Clearly, the larger the spare capacity, the higher the capability of the network to react to failures and other unexpected events. Moving towards the right side of the graph, we first meet the *shared-classic* and the *dedicated-classic* strategies, and finally the two *robust plus dedicated* protection. Classic schemes are considered more conservative than smart ones because all backup capacity is kept activated. It is worth to notice that, in term of consumption, switching off the backup links with dedicated protection is more

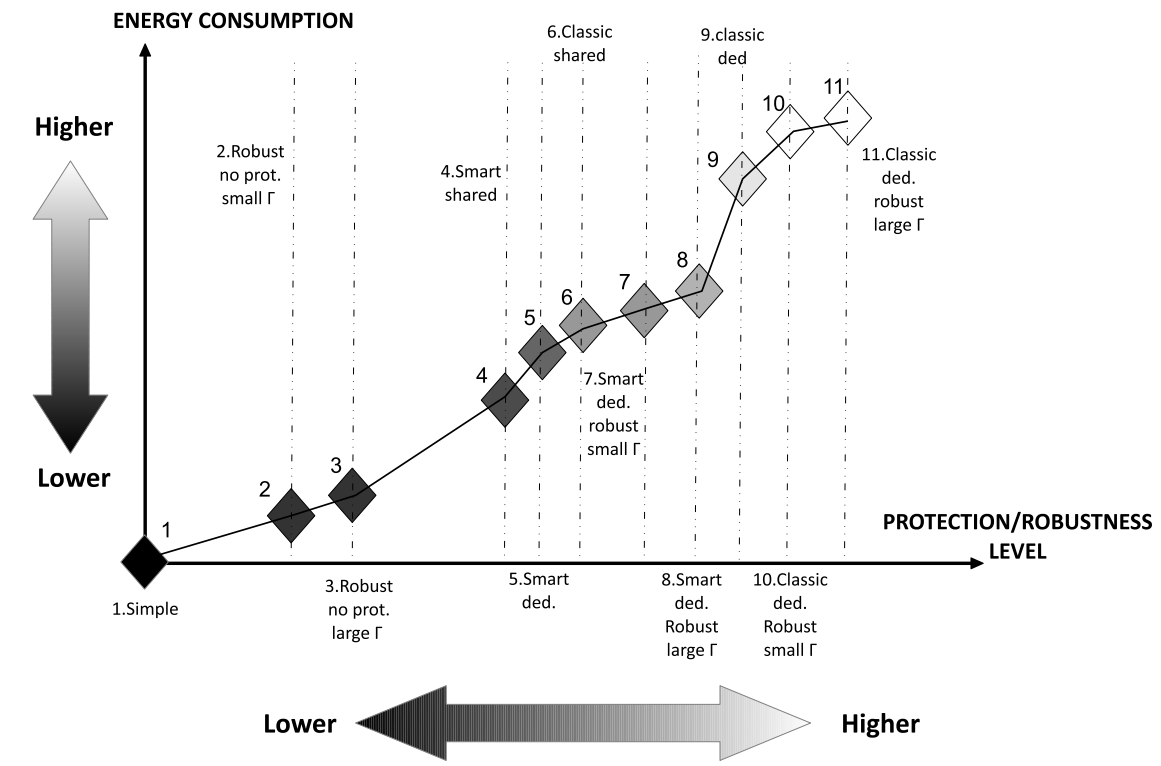


Figure 4.3: Savings vs Protection/Robustness.

efficient than the complex shared protection scheme (see detailed results in following subsections).

We considered now twelve instances associated to the smallest network, `polska`, and detailed results on them. We solved the MILP formulation of each problem with a time limit of one hour. The effectiveness of the computing methods, i.e. computing times, solution optimality, absolute savings, are evaluated, as well.

Protection strategies: energy efficiency

Results concerning the protection scheme models are reported in Tables 4.5-4.7. In Table 4.5, the energy savings achieved by *simple*, *dedicated-classic*, and *shared-classic* models are shown. Column $\%E_c$ represents the ratio between the energy consumption of the optimized versus the full active network. Column gap_{opt} represents the gap of the final solution w.r.t. to the best lower bound computed by CPLEX. Column gap_{simple} represents the relative increase of energy consumption due to the survivability requirement: it is computed as $\%E_c^{prot} - \%E_c^{simple}/\%E_c^{simple}$, where $\%E_c^{prot}$ and $\%E_c^{simple}$ represent the energy consumption of the optimized network w.r.t. the full active one, for the unprotected and protected case, respectively. As expected, the explicit imple-

Table 4.5: Comparison between simple and protected solutions obtained by solving the exact model with 1h time limit on `polska` instances.

	Polska - Exact model							
	simple case		***dedicated prot classic***			***shared prot classic***		
ID	$\%E_c$	gap_{opt}	$\%E_c$	gap_{opt}	gap_{simple}	$\%E_c$	gap_{opt}	gap_{simple}
$alfa_1$	60,6%	1,3%	71,4%	1,4%	17,8%	66,9%	3,6%	10,3%
$alfa_2$	60,5%	0,9%	71,3%	0,9%	17,8%	66,3%	4,4%	9,6%
$alfa_3$	60,3%	0,6%	71,4%	0,7%	18,4%	70,4%	8,9%	16,7%
$delta_1$	50,7%	2,4%	62,2%	2,6%	22,7%	59,3%	10,1%	17,0%
$delta_2$	50,1%	0,8%	61,4%	3,3%	22,7%	60,3%	15,5%	20,5%
$delta_3$	50,3%	0,4%	61,7%	2,7%	22,6%	61,7%	15,8%	22,6%
$delta_1$	60,0%	1,4%	70,9%	0,9%	18,1%	66,2%	3,2%	10,3%
$delta_2$	59,8%	0,7%	70,7%	0,8%	18,1%	65,7%	3,6%	9,8%
$delta_3$	59,7%	0,0%	70,8%	0,5%	18,6%	70,9%	11,1%	18,8%

mentation of a protection scheme increases the network energy consumption, in fact the energy-aware approaches keep activated additional resources to cope with possible failures. In the case without protection, the consumption E_c varies from 50.1% to 60.6%, in the *dedicated-classic* case network consumption is between 61.4% and 71.4%, with absolute and relative increase, on average, of 10% and 20%, respectively. By considering the more sophisticated *shared-classic* protection, the consumption can be reduced, w.r.t. the *dedicated-classic* case, up to 5%. However, while for the *dedicated-classic*, the model computes nearly optimal solutions within the time limit of one hour (gap_{opt} usually lower than 1% and never above 3.5%), for the *shared-classic* case the gap from the best lower bound is in some instances larger than 15% (Instances 6-7), as the model is more complex and requires a high computational effort. For this reason, in some instances the reported difference between shared and dedicated protection consumption is smaller than 1% (Instances 6-7-11).

To overcome this problem, we applied what we called the single time period heuristic (STPH). The central idea is to solve the problem for each given time period on a modified model to take into account connecting constraints related to limiting the switching on-off of the devices (see [J12]).

Heuristic results are reported in Tables 4.6 and 4.7. In Table 4.6 we analyze the gap between exact model and STPH solutions. Columns $Heur_{gap}$ represent the difference between the energy consumption obtained by the model and that achieved by STPH, i.e. $E_c^{heur} - E_c^{model}$. In Table 4.7 we compare the saving improvement achieved by the smart protection solution produced by STPH w.r.t. the classic one. Columns $\Delta_{smart}^{classic}$ represent the absolute difference between the energy consumption obtained with the smart and the classic models. The time limits are reported, as well: in Tables 4.6 and 4.7, differently from Table 4.5, TL represent the time limit imposed to CPLEX when solving a single

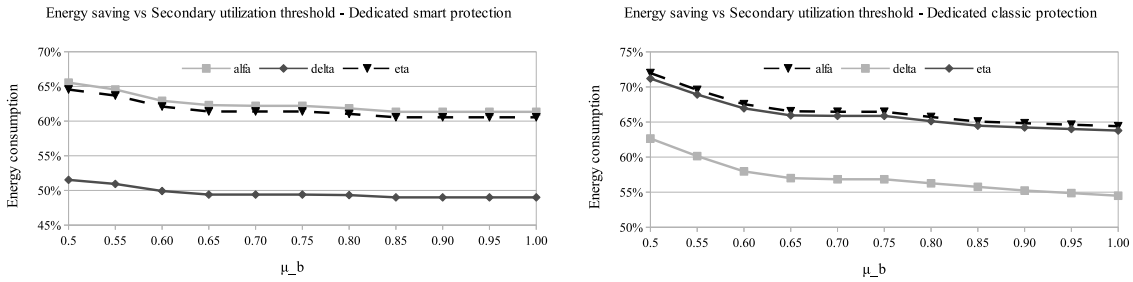


Figure 4.4: Analysis of the trade-off between energy savings and network congestion, obtained by adjusting the secondary utilization threshold $\bar{\mu}$ (μ_b in the figure) from 0.5 to 1 when solving STPH.

time period of STPH.

Table 4.6 shows that, by using STPH with a time limit of 6 minutes, we reduce the energy consumption of the solutions with shared protection up to about 5% (Instances 3-6-7-11). The difference between shared and dedicated protection for the instances for which the gap obtained solving the model is large (Instances 6-7-11) is therefore increased. Furthermore, it is worth pointing out that, even for the instances of the dedicated case solved at optimality or with a very small gap using the complete model, the gap between STPH and the formulation is very small, varying between 0.6% and -0.2%, negative values meaning that STPH solutions improve upon the sub-optimal solutions found by CPLEX when solving the model. Having shown the good quality of STPH algorithm solutions in the remainder of this section we report only the results obtained by solving STPH, for practical and space reasons.

Table 4.6: Comparison between the energy saving achieved by solving the exact model (with 1h TL) and running the single time period heuristic with different types of protection, `polska` instances.

	Polska - Exact model vs STPH					
	simple case		***dedicated prot classic***		***shared prot classic***	
ID	$Heur_{gap}$	TL	$Heur_{gap}$	TL	$Heur_{gap}$	TL
$alfa_1$	0,00%	60s	0,1%	30s	-0,3%	360s
$alfa_2$	0,25%	60s	0,1%	30s	-0,3%	360s
$alfa_3$	0,16%	60s	0,0%	30s	-4,0%	360s
$delta_1$	0,41%	60s	0,6%	30s	-2,1%	360s
$delta_2$	0,00%	60s	-0,1%	30s	-3,6%	360s
$delta_3$	0,28%	60s	-0,2%	30s	-4,5%	360s
eta_1	0,28%	60s	0,1%	30s	-0,3%	360s
eta_2	0,28%	60s	0,2%	30s	-0,4%	360s
eta_3	0,17%	60s	0,1%	30s	-4,9%	360s

The possibility of putting to sleep the line cards carrying only the backup links (*smart* protection) is expected to substantially decrease the energy consumption of the network w.r.t. the *classic* case. This hypothesis is clearly confirmed by the results of Table 4.7, where we observe that *smart* protection allows to reduce the consumption of the protected solutions (w.r.t. the total network consumption) by up to 7.1% and 3.9%, for the dedicated and shared case, respectively. *Smart shared* produces smaller energy consumption reduction, w.r.t. the non smart case, than *smart dedicated*. The smaller improvement obtained by the smart scheme with shared protection w.r.t. dedicated protection is explained by the smaller amount of backup capacity that can be put to sleep. The most important result that is worth pointing out here is that with the smart scheme, dedicated protection can be more energy efficient than classic shared protection, while being less computationally expensive and easier to implement.

Protection strategies: congestion analysis

Concerning the congestion, it is necessary to remind that shared protection, due to the high efficiency of the backup allocation scheme, can deal with levels of traffic that cannot be managed by the dedicated protection scheme, without violating the maximum utilization constraints. To compute $\varpi_{50\%}^{85\%}$, shared instead of dedicated protection can be used. If shared protection is chosen, this value increases up to 25%. Therefore, although the computational effort required to handle shared protection is significantly higher than in the dedicated case, shared protection is worth being implemented to further reduce network congestion.

Table 4.7: Comparison between the energy saving achieved by STPH with classic and smart protection schemes, `polska` instances.

	Polska - STPH - Classic vs Smart			
	dedicated		***shared***	
ID	$\Delta_{smart}^{classic}$	TL	$\Delta_{smart}^{classic}$	TL
$alfa_1$	-3,9%	30s	-1,9%	360s
$alfa_2$	-3,5%	30s	-2,2%	360s
$alfa_3$	-3,2%	30s	-1,9%	360s
$delta_1$	-7,1%	30s	-3,4%	360s
$delta_2$	-5,9%	30s	-3,9%	360s
$delta_3$	-5,5%	30s	-3,6%	360s
eta_1	-4,2%	30s	-2,0%	360s
eta_2	-3,8%	30s	-2,3%	360s
eta_3	-3,5%	30s	-2,1%	360s

To better understand the balance between network congestion and energy savings, we report in Figure 4.4 the network energy-consumption computed by varying the secondary maximum utilization threshold $\bar{\mu}$ from 0.5 to 1. In this specific set of tests, we considered

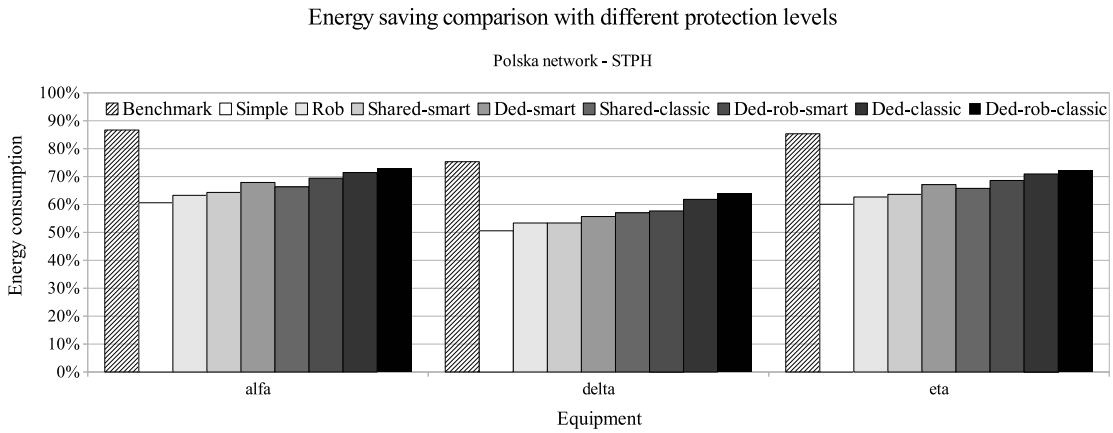


Figure 4.5: Energy savings achieved by STPH when implementing the different protection schemes on `polska` instances.

dedicated protection and traffic matrices obtained by using $\varpi_{50\%}^{50\%}$ instead of the $\varpi_{50\%}^{85\%}$. In fact, with $\varpi_{50\%}^{85\%}$ the problem would not be feasible in case of $\mu_b < 0.85$. Figure 4.4 shows that the difference between the network consumption obtained with $\mu_b = 0.5$ and $\mu_b = 1$ varies from 4% to 8%. The plot clearly shows how a network provider can balance energy savings and network congestion according to his own requirements.

To conclude this section, we report a compact view of the energy savings obtained with the different protection strategies, where as for all other results, the benchmark is the full active network (see Figure 4.5).

Largest networks

In the second group of tests, we experimented with `nobel-germany`, `nobel-eu` and `germany` network by running STPH or STPH-RP (the restricted path version of STPH, where the possible routing of each demand is reduced to a set of pre-calculated paths, see again [J12] for details). For comparison purposes, a restricted set of instances were tested with both procedures. STPH-RP was then used to solve instances that were too computationally demanding to be efficiently solved in a reasonable amount of time by the simple STPH. The time limit for the single time period used to run STPH and STPH-RT are reported in Table 4.8. If a method cannot provide a feasible solution for an instance a "/" is reported in the corresponding entry of the table (only STPH can provide feasible solutions for instance `nobel-germany`).

Figure 4.6 reports the results for the `nobel-germany` network, while Figure 4.7 reports the results for the `nobel-eu` network. First of all, we can observe that the consumption trend obtained for larger instance is similar with the one obtained with polska (see Figure 4.5). The only difference that can be observed is that the energy consumption for the *dedicated-smart* case is on average smaller than the one of the *shared-classic* case.

This can be explained as the solver is not able to efficiently solve the shared protection model, even for the single period, and to obtain a small gap when the instance dimensions increase. It is worth noting that, in some tests, the solution computed by the warm start procedure cannot be improved by the solver within the chosen time-limit. Besides, due to memory limits (8GB of RAM), the shared protection instances could not even be initialized for the `nobel-eu` and `germany` networks. Thus, as a solution feasible for the dedicated problem is naturally feasible for the shared one, the solutions obtained by solving the *dedicated* problem are applied also for the shared case.

Finally, to further confirm the good performance of STPH-RP, the average network consumption values computed on the `germany` network with all the different protection schemes are reported in Figure 4.9. Also in this case the heuristic method provides significant savings, obtaining final network consumption from 60% up to 80% of the original value.

Table 4.8: CPLEX time limits for the single time period to solve `nobel-germany`, `nobel-eu` and `germany` instances with different types of protection.

	simple		*dedicated*		*shared*	
Net	TL_{STPH}	$TL_{STPH-RP}$	TL_{STPH}	$TL_{STPH-RP}$	TL_{STPH}	$TL_{STPH-RP}$
nobel-ger	60s	/	90s	/	360s	/
nobel-eu	300s	/	300s	300s	/	/
germany	/	600s	/	600s	/	/

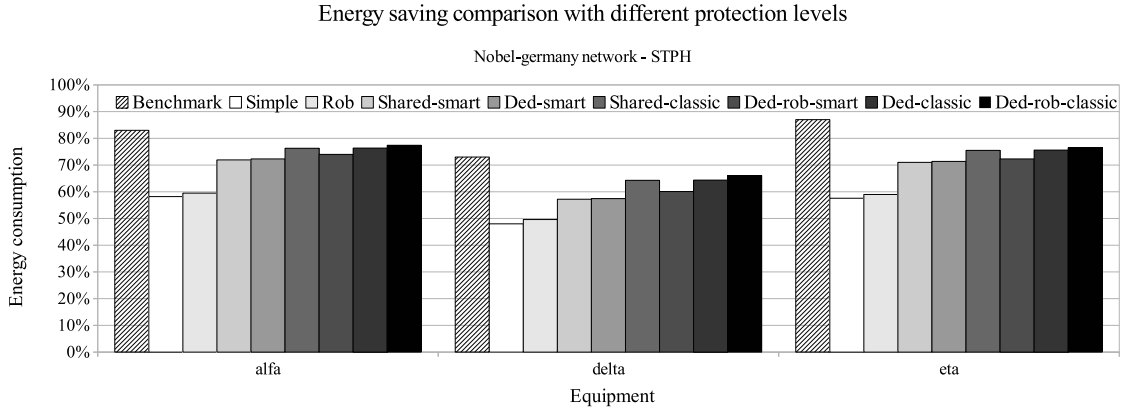


Figure 4.6: Energy savings achieved by STPH when implementing the different protection schemes on `nobel-germany` instances.

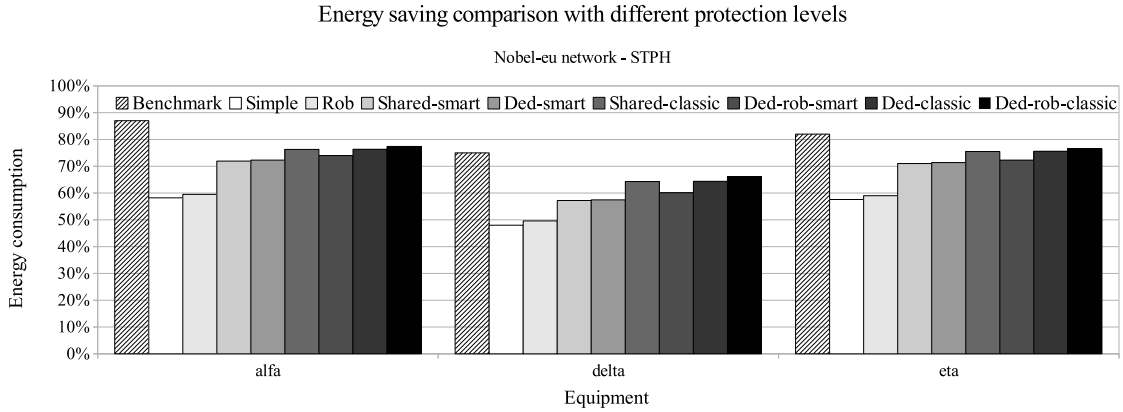


Figure 4.7: Energy savings achieved by STPH when implementing the different protection schemes on `nobel-eu` instances.

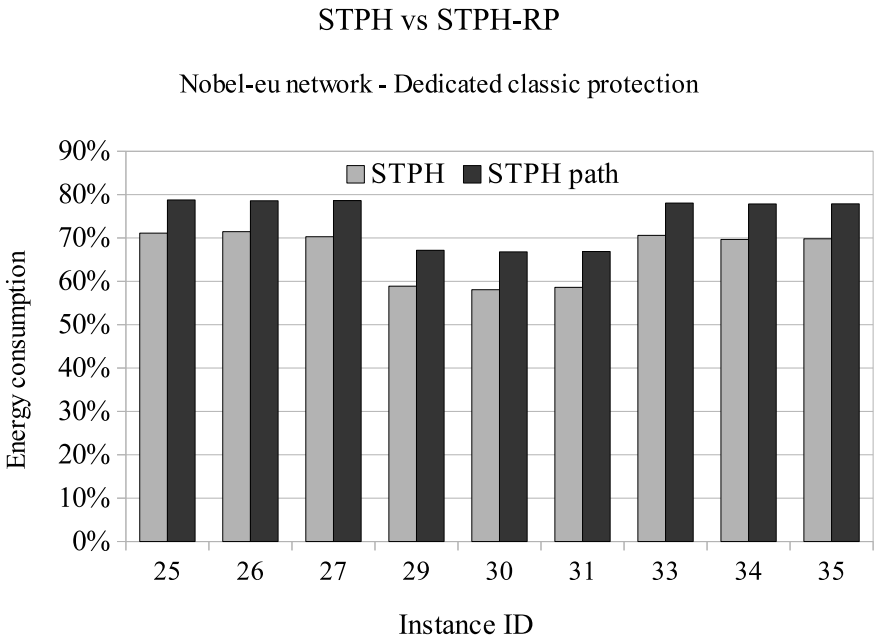


Figure 4.8: Energy saving comparison between STPH and STPH-RP on `nobel-eu` network with *dedicated classic* protection.

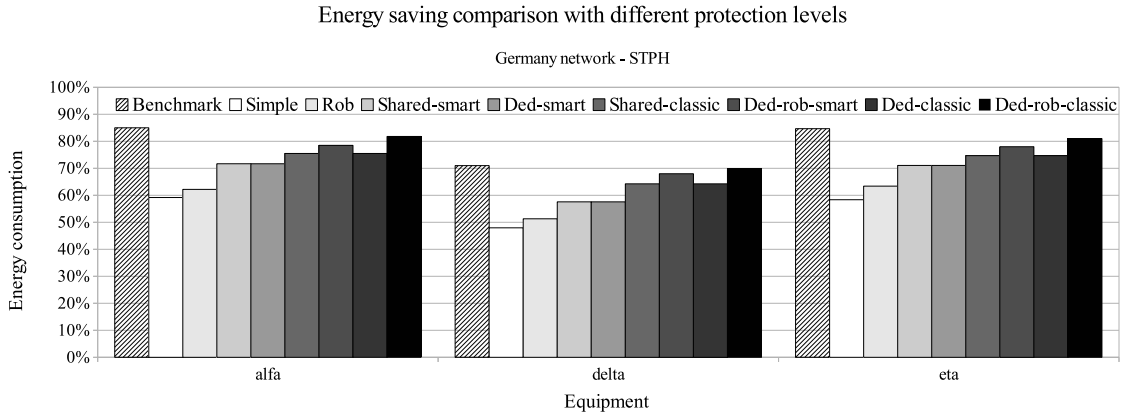


Figure 4.9: Energy savings achieved by STPH-RP when implementing the different protection schemes on `germany` instances.

4.3 Smart protection: refining the solution method

Even if the IP-BEANM problem with integer routing even in its simplest version (single period, no protection) is NP-hard (by reduction from the Directed Two-commodity Integral Flow (DTIF), problem ND38 in [80]), the computational effort necessary to solve it with state of the art solvers is acceptable for planning strategies on medium size networks, as it was shown in the previous section. Even adding the dedicated protection the computational time is still acceptable, but when shared protection is considered the problem becomes computational challenging, even in its single period version (as we observed by our experiments). Therefore, we decided to investigate the shared protection case and its properties with the aim of proposing enhanced optimization methods to solve this problem. We started with a simplified version of the problem where a single period is considered, the possibility to switch on and off nodes is neglected (therefore node consumption is a constant term) and a single card is considered ($n_{ij} = 1$). In this section, we report a summary of the methodology and the results we obtained (see [J5, J4] for a detailed description and the complete results).

Differently from the previous models, here we represent the network with an indirected graph $G(V, E)$, as we assumed that energy is associated to edges (that is equivalent to impose constraint $w_{ij} = w_{ji}$ as we did in [J13]), therefore link activation variables w_{ij}, w_{ji} will be replace by edge activation variables w_e. As the model used in [J5, J4] is slightly different from the ones presented until now and to allow the reading of the section "quite independently" from the previous ones, we report here what we called the Compact Formulation (to distinguish it from our proposed Projected Formulation):

$$\text{(CF)} \quad \min \sum_{e \in E} \pi_e w_e$$

$$\sum_{\substack{j \in V: \\ (i,j) \in A}} x^d_{ij} - \sum_{\substack{j \in V: \\ (j,i) \in A}} x^d_{ji} = \begin{cases} 1 & \text{if } i = o_d, \\ -1 & \text{if } i = t_d, \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V,\, d \in D \tag{4.33}$$

$$\sum_{\substack{j \in V: \\ (i,j) \in A}} \xi^d_{ij} - \sum_{\substack{j \in V: \\ (j,i) \in A}} \xi^d_{ji} = \begin{cases} 1 & \text{if } i = o_d, \\ -1 & \text{if } i = t_d, \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V,\, d \in D \tag{4.34}$$

$$\sum_{d \in D} r^d (x^d_{ij} + x^d_{ji}) \leq U w_e \qquad \forall e = \{i, j\} \in E \tag{4.35}$$

$$\sum_{d \in D} r^d (x^d_{ij} + x^d_{ji} + g^d_{et}) \leq U \qquad \begin{array}{l} \forall e = \{i, j\} \in E, \\ \forall t = \{u, v\} \in F \\ e \neq t \end{array} \tag{4.36}$$

$$g^d_{et} \geq x^d_{uv} + x^d_{vu} + \xi^d_{ij} + \xi^d_{ji} - 1 \qquad \forall e = \{i,j\}, t = \{u,v\} \in F \tag{4.37}$$

$$x^d_{ij} + x^d_{ji} + \xi^d_{ij} + \xi^d_{ji} \leq 1 \qquad \forall e = \{i,j\} \in E \tag{4.38}$$

$$x, \xi \in \{0,1\}^{2\times|D|\times|E|}, w \in \{0,1\}^{|E|}$$

Parameter U represents the edge bidirectional bandwidth capacity ($U = \mu_{ij}c_{ij}$). F is the set of edges subject to failure, in the majority of the cases F corresponds to E, but we decided to use a separated set to have a more general model and allow to put in evidence the impact of edge failures. Constraints (4.33) and (4.34) are the flow balance constraints. Capacity constraints (4.35) ensure that primary flows do not exceed the capacity on an edge (that is zero if the edge is not active). Constraints (4.38) impose that the primary and the secondary paths are edge-disjoint and that each edge is crossed at most once by any commodity. The shared capacity constraints (4.36 and 4.37) guarantee that, in case of failure, the primary and the secondary flows do not exceed the maximal edge capacity (obtained activating the edge).

The Compact Formulation (CF) is very challenging to solve in practice. Even solving the LP relaxation may be highly time consuming and it provides a poor bound for the integer problem. Just to give some order of magnitude, at the root node we obtained on average 40% of gap in $400s$ on networks with 15 nodes and 30 demands (with a worst case time of 1700s). We applied existing valid inequalities and proposed new ones for strengthening the CF. Furthermore, we proposed a projected formulation. We compared the original formulation, the formulation with the additional cuts and the projected formulation. We proved that the projected formulation produces the best results in terms of computational bound and time. In the following, we present briefly the added cuts and the projected formulation. We conclude the section with a short summary of the results.

Many classes of valid inequalities have been studied both for unstructured binary problems and general network design problems. In the following, we show that some of the known cuts can be directly applied to our problem, others can be adapted to the problem and additional problem-related cuts can be derived.

CutSet (CS) inequalities Let $\{P : V \setminus P\}$ be a partition of the nodes, let $D(P) \subseteq D$ be the demands having source and destination separated by the cut and let $\delta(P)$ be the set of edges having endpoints in different subsets of the partition, then the active edges in $\delta(P)$ must have enough capacity to support the routing of the primary paths of the demands in $D(P)$:

$$\sum_{e\in\delta(P)} w_e \geq \left\lceil \frac{\sum_{d\in D(P)} r^d}{U} \right\rceil \qquad \forall P \subseteq V \tag{4.39}$$

We can observe that, differently from the classic form of cutset inequalities for survivable networks ([90, 122]) where the capacity of the cut depends on all the edges in the cut, here only the no-fault scenario requires edge activation and therefore only active edges affect the cut.

Demand Lower Bound (DLB) Although no edge activation is needed for the secondary paths, they contribute to the capacity consumption, therefore we can estimate a lower bound of the total demand traversing an edge and impose the following VI:

$$\max_{h \in D}\{r^h(\xi^h_{ij} + \xi^h_{ji})\} + \sum_{d \in D} r^d(x^d_{ij} + x^d_{ji}) \leq U \quad \forall e = \{i,j\} \in E \tag{4.40}$$

The same argument has been used in [90] to introduce lower bounds on the traffic across a cut, that cannot be used here, due to smart protection. Contrary to shared capacity constraints (4.36), inequalities (4.40) are fault independent. Moreover, they are polynomially many ($|E| \times |D|$).

Variable Activation (VA) That connect routing and edge activation variables:

$$x^d_{ij} + x^d_{ji} \leq w_e \quad \forall e = \{i,j\} \in E, d \in D \tag{4.41}$$

Energy Cover (EC) Combining cover inequalities and edge activation variables, we obtain:

$$\sum_{d \in C}(x^d_{ij} + x^d_{ji}) \leq (|C| - 1)w_e \quad \forall e = \{i,j\} \in E, C \text{ demand cover} \tag{4.42}$$

4.3.1 The projected formulation

Projecting out some of the variables is a well-known technique, which is used to solve many network design problems [18, 26, 30, 35, 121, 122, 155]. Since the energy consumption only depends on the primary paths, a natural choice is to project out secondary path variables $\boldsymbol{\xi}$ and the related constraints, obtaining the Projected Formulation below:

$$(PF) \quad \min \sum_{e \in E} \pi_e w_e$$

$$\sum_{\substack{j \in V: \\ (i,j) \in A}} x^d_{ij} - \sum_{\substack{j \in V: \\ (j,i) \in A}} x^d_{ji} = \begin{cases} 1 & \text{if } i = o_d, \\ -1 & \text{if } i = t_d, \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V,\ d \in D \tag{4.43}$$

$$\sum_{d \in D} r^d(x^d_{ij} + x^d_{ji}) \leq U w_e \qquad \forall e = \{i,j\} \in E \tag{4.44}$$

$$x^d_{ij} + x^d_{ji} \leq 1 \qquad \forall e = \{i,j\} \in E \tag{4.45}$$

$$x \in \{0,1\}^{2 \times |D| \times |E|}, w \in \{0,1\}^{|E|}$$

For the sake of brevity, a pair (x, f) is said *coherent* if it is feasible for the continuous relaxation of PF. In [J4], we showed that, unlike for classical sustainable network design problems, feasibility can be ensured working only on flow variables f, as edge activation variables x have no impact on the existence of a set of secondary paths. Let consider a current flow solution $\bar{f}$, two different set of inequalities can be added to force feasibility for the original model, depending on considering the continuous relaxation or the integer version of CF.

On the continous relaxation:

$$\sum_{k\in K}\sum_{e=\{i,j\}\in E}\left(d_k\sum_{t\in F}\mu_e^t+\tau_e^k\right)(f_{ij}^k+f_{ji}^k)\leq\sum_{k\in K}\ell_{t_k}^k+\sum_{e=\{i,j\}\in E}\left(\sum_{k\in K}\tau_e^k+\sum_{t\in F}U\right) \tag{4.46}$$

for any $\mu_e^t \geq 0$, $\tau_e^k \geq 0$ and $\ell_{t_k}^k$ being the shortest path length from s_k to t_k according to weights $w_e^k = \sum_{t=\{u,v\}\in F} d_k(\bar{f}_{uv}^k + \bar{f}_{vu}^k)\mu_e^t + \tau_e^k$. We note that, since backup paths must be fault-independent, it is not possible to check one failure scenario at a time, but it is needed to find unique secondary paths $\boldsymbol{\xi}$ that are feasible for all the scenarios at the same time.

On the integer version:

$$\sum_{k\in K}\sum_{e=\{i,j\}\in E_{\bar{\mathbf{f}}}^k}\left(f_{ij}^k+f_{ji}^k\right)\leq -1+|E_{\bar{\mathbf{f}}}| \tag{4.47}$$

where $E_{\bar{\mathbf{f}}}^k$ is the set of edges used by the primary path of k according to $\bar{\mathbf{f}}$ and let $E_{\bar{\mathbf{f}}} = \sum_{k\in K}|E_{\bar{\mathbf{f}}}^k|$. Let $\mathcal{F}(\bar{\mathbf{f}})$ be the set of solutions $\mathbf{f}$ such that $E_{\bar{\mathbf{f}}}^k \subseteq E_{\mathbf{f}}^k$. Note that the above definition is not affected by the direction in which edges are used, but only by their use, in one direction or the other.

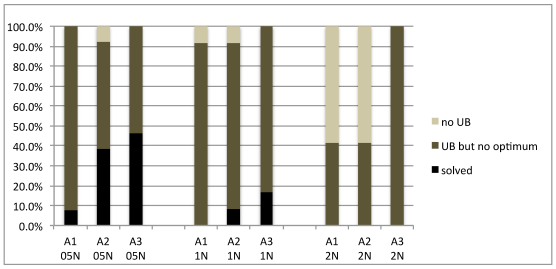
4.3.2 Short analysis of the computational results

We compare three solution strategies: complete formulation solved by CPLEX used as stand-alone solver (A1), complete formulation with cut generation (A2), projected formulation (A3). For the complete formulation (A2) we add multiple cuts every time, while for the projected formulation (A3) we add only one cut at a time. The reason is that solving the LP relaxation of the complete formulation is itself time consuming. Therefore we add more cuts at each iteration in the hope of cutting, together with the current solution, other fractional solutions as well, instead of generating them in successive iterations, thus reducing the number of times LP relaxation is solved. Single node CS inequalities are added to the initial formulation of both A2 and A3 (CS inequalities are first separated heuristically and only if no violated inequality is found by the heuristic approach, we proceed with exact separation). In the branch-and-cut algorithm, for both A2 and A3, we generate CS, VA, and EC at every node of the branch-and-bound tree. Furthermore, for A2 we generate DLB, while for A3 we need inequalities necessary to recover feasibility of the projected formulation (let call them FPP). VA and DLB are separated by lookup, while CS, EC and FPP are separated exactly solving an optimization problem.

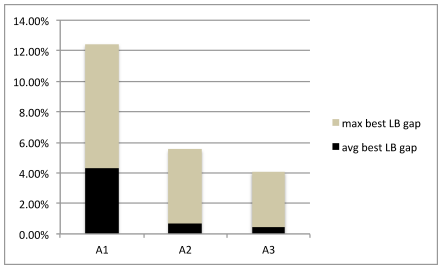
We consider instances with 15, 20 and 25 nodes. The instances with 15 nodes are the test bed for comparing the three approaches. The results proved that the projected formulation outperforms the complete one (A1 and A2), we used the second test bed, including instances with 20 and 25 nodes, to assess its scalability.

The network topologies have been generated using Georgia Tech Internetwork Topology Models code [88]. The number of edges varies in $(2|N|, 4|N|)$. Edge capacity has

been set so as to guarantee that there exists no feasible solution for the dedicated case. Edge density is chosen in $\{0.2, 0.4, 0.6\}$, and it represents the probability of accepting an edge during the graph generation, therefore an edge density of 0.6 means that there are on average $0.6|N|(|N|-1)/2$ edges. The set of demands have been randomly generated and its cardinality is chosen proportional to the number of nodes $\{0.5|N|, |N|, 2|N|\}$. The experiments are made on an 4× Intel Core i5 @ 3.20GHz using CPLEX 12.6 with single thread option and one hour time limit.



(a) Percentage of solved instances, of upper bound found and of unresolved instances

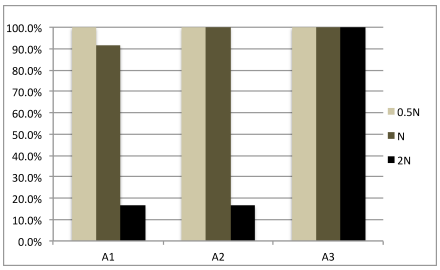


(b) Average and maximum gap with respect to the best lower bound

Figure 4.10: Results at root node for instances with 15 nodes

Figure 4.11a gives the number of instances solved to optimality with different number of demands. A3 clearly outperforms A1 and A2. In fact, it solves to optimality all the instances. Indeed, also A2 can solve to optimality all the instances with $|K| = 0.5|N|$ and $|K| = |N|$, while A1 can solve to optimality all the instances with $|K| = 0.5|N|$ demands but cannot solve one instance with $|K| = |N|$. A1 and A2 can solve only 2 out of 12 instances with $|K| = 2|N|$ demands, and provide an average gap of about 20% on the others. A1 and A2 require higher computational times then A3: A2 requires about 100 times the CPU time required by A3.

Having proved that the projected formulation outperforms the others, we then consider larger instances. In instances, in Figure 4.12 are reported results for instances with 20 and 25 nodes, that show that A3 behaves rather well also on larger instances. The final gaps are not dramatic: for the 20 node instances it is always below 8%, while it is always below 11% for the 25 node instances. In order to further investigate the impact of the number of demands, we consider an additional set of instances with 20 nodes and about 70 edges, with $3|N|$ and $4|N|$ demands. We observe that the final gap for the instances with 20 nodes and $3|N|$ demands is comparable with the one obtained for instances with 25 nodes and $2|N|$ demands. To summarize: the projected formulation outperforms the complete one in terms of quality of results and computational time. Indeed, the shared capacity constraints are very heavy to dealt with, as it is shown by the computational tests on the complete formulation and projecting them out produce very good results.



(a) Percentage of instances solved to optimality

$\|K\|$		A1		A2		A3	
		gap	time	gap	time	gap	time
0.5N	avg	0.0%	22.5	0.0%	13.4	0.0%	2.3
	max	0.0%	152.0	0.0%	31.0	0.0%	4.0
1N	avg	0.3%	562.8	0.0%	635.5	0.0%	7.9
	max	3.2%	TL	0.0%	2813.0	0.0%	17.0
2N	avg	18.9%	3113.1	19.0%	3164.5	0.0%	245.1
	max	34.4%	TL	54.1%	TL	0.0%	552.0

(b) Gap and Solving time

Figure 4.11: Results B&B for instances with 15 nodes

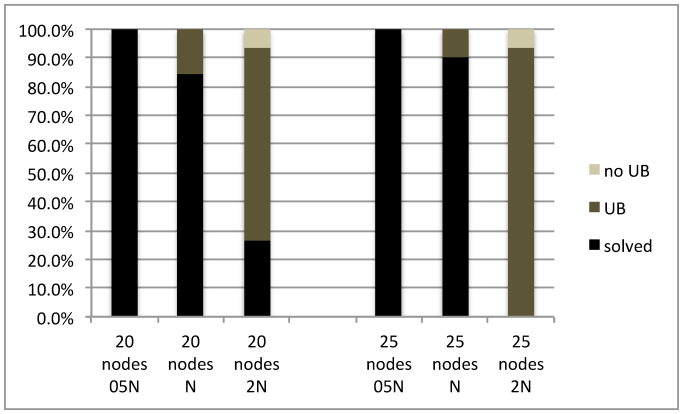


Figure 4.12: Percentage of solved instances, of upper bound found and of unresolved instances for the 20 and 25 node instances - A3

4.4 Conclusions

In this chapter, the main contributions on energy-aware telecommunication networks are summarized. The focus is on energy-aware network management for IP-networks, i.e. the routing and the dynamical switching on/off of the devices to allow energy consumption reduction without affecting the requested QoS. On this problem, the contributions are of different nature: first of all, a comprehensive literature review, in term of mathematical programming models is proposed, allowing to show where the models existing in the literature introduce some novelty with respect to "classical" network management, and when they were just a reinterpretation in the "energy-aware" perspective of well known models; second, an analysis on the impact of different protection strategies is reported in term of energy consumption and congestion; last, but not least, inspired by this application problems, and driven by the need of more effective algorithms for solving models considering shared-protection [6] a new solution algorithm based on a projected formulation is proposed and its effectiveness with respect to the current state of the art is showed.

[6]The mechanism that, in its "smart"-version showed the better performances in terms of energy consumption

5 The integration of telecommunication networks and computing systems

Even if computing and networking components have been designed and managed quite independently so far, the current trend is to have them more strongly integrated for improving performance and efficiency of Cloud services offered to end users [15]. The integration of computing and networking components can be used not only to provide service flexibility to end users (as for example, in the Virtual Network Function paradigm, as described in Section 5.2), but also to manage in a flexible way resources available in geographically-distributed computing centers and the network interconnecting them, as we will show in Section 5.1.

5.1 Network and service centers joint management: a green-perspective

A key enabler of Cloud/network cooperation is the use of geographically distributed service centers. Distributed Cloud service provisioning allows to better balance the traffic/computing workload and to optimize energy consumption by exploiting load variations and energy cost variations over time in different locations.

To assess the potential savings that can be achieved with the joint management of service centers and the network connecting them and to highlight the relevant parameters that impact on the overall system performance, we develop a MILP (Mixed Integer Linear Programming) model to describe the problem. The resource utilization and load allocation scheduling is performed on a daily basis assuming a central decision point and the availability of traffic patterns for different time periods (as in [179],[41],[13],[J16]).

The model optimizes the load allocation to a set of geographically distributed service centers (SCs) where virtual machines (VMs) are assigned to physical servers in order to serve requests belonging to different classes. The goal is to minimize the total energy cost considering the time-varying nature of energy costs and the availability of green energy at different locations. The traffic can be routed to SCs using a geographical network whose capacity constraints and energy consumption are accounted for. The network is represented in an aggretated form considering virtual paths connecting the SCs (an extented version with a more refined network model was also develped, the interested reader can refer to [J13]). In Figure 5.1, a schematic view of the system is reported.

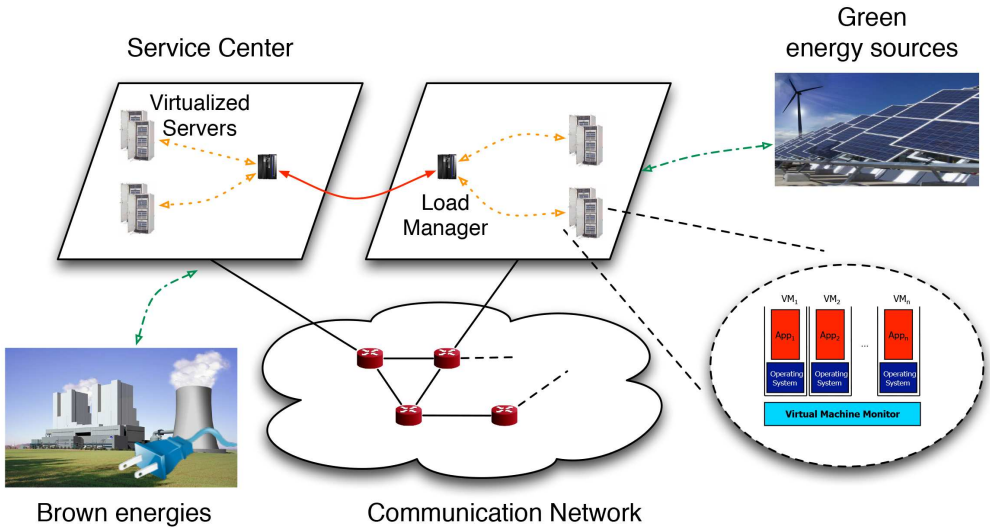


Figure 5.1: Reference Cloud System Architecture.

5.1.1 Service Centers and Communication Network joint management

To the reader convenience, the definitions of sets and parameters are summarized in Tables 5.1-5.2 (and Table 5.4 for energy related ones) and the decision variables are summarized in Table 5.3.

We denote with $\mathcal{N}$ the set of SCs. Each SC has a set of different types of VMs $\mathcal{L}$ that can serve incoming requests. Among the many resources, we focus on the CPU and bandwidth as representative resources for the resource allocation problem, consistently with [149, 157, 39, 36, 37]. VM and SC have capacity limits: P_{il} is the capacity of a VM of type l at the SC i, while C_i is the overall SC i computing capacity.

The hosted services can be heterogeneous with respect to computing demands, workload intensities, and bandwidth requirements. Services with different computing and workload profiles are categorized into independent request classes, where the system overall serves a set $\mathcal{K}$ of request classes. Different request classes need different kinds of VMs and have different bandwidth and CPU requirements. Three parameters are used to model such features. Parameter m_{kl} is equal to 1 if request of class k can be served by a VM of type l, parameter b_k represents the bandwidth requirement of class k requests, while D_k represents the overall CPU demanding time [66] for serving a request of class k on a VM of capacity 1.

Finally, the planning horizon of 24 hours is discretized in time bands that are represented by set $\mathcal{T}$.

Each SC is characterized by a specific traffic profile for each time band: the local arrival rate for requests of class $k \in \mathcal{K}$ at the SC $i \in \mathcal{N}$ at time $t \in \mathcal{T}$ is denoted with λ_{ik}^t and we assume that the workload profile is periodic [27]. Nevertheless, differently from the works on EANM, we do not use a periodic model, because we integrate green energy resources (eolic, solar, etc), that for their nature are very variable. Therefore, to

$\mathcal{N}$	set of Service Centers
$\mathcal{L}$	set of VM types
$\mathcal{K}$	set of request classes
$\mathcal{T}$	set of time bands

Table 5.1: Sets definition

take into account the change in states of devices, we introduce parameters representing the state of the system at time zero.

Concerning the network, to estimate the energy consumption, we consider the number of physical hops R_{ij} and a capacity Q_{ij} for each path connecting each pair of SCs i and j.

Requests parameters

λ_{ik}^t	Local incoming workload arrival rate for request class k at SC i at time band t (Req/hour)
m_{kl}	VM requirement parameter: equal to 1 if class $k \in \mathcal{K}$ can be served by type l VM, 0 otherwise
b_k	bandwidth requirement for request class k
D_k	needed time to serve request class k on a VM of capacity 1

VM and SC capacities

P_{il}	capacity of a VM of type l at SC i
C_i	SC i overall computing capacity
$\overline{\mathcal{U}}$	average VMs utilization

Network capacities

R_{ij}	total number of routers in the link (i, j)
Q_{ij}	maximum bandwidth available on path (i, j)

Status (time 0) parameters

w_{il}^0	number of active VMs of type l in SC i at time 0
z_{ij}^0	link (i, j) status at time 0

Table 5.2: Parameters. Part 1 (Requests, capacities and status parameters)

Before introducing the parameters concerning energy, we present the main contraints of the model.

Continuous non negative variable x_{ijkl}^t represents the arrival rate of requests of class k at SC i, which are served in SC j by a VM of type l, at time t. In other terms, x_{ijkl}^t variables represent the optimal partition between forwarding or in-site serving. Note that x_{iikl}^t indicates the request rate that is originated and served locally.

Integer variable w_{il}^t represents the number of VMs of type l used in SC i at time band

Continuous non-negative variables

x_{ijkl}^t	Arrival rate for class k request redirected from SC i to SC j served with a type l VM
y_i^t	Green energy used in SC i at time t

Integer variables

w_{il}^t	Number of type l VMs running in SC i at time band t
$\overline{w}_{il}^t$	Number of VMs of type l turned on with respect to time $t-1$ in SC i
$\underline{w}_{il}^t$	Number of VMs of type l turned off with respect to time $t-1$ in SC i

Binary variables

z_{ij}^t	Whether the link (i,j) is active in time band t (binary)
$\overline{z}_{ij}^t$	Whether the link (i,j) has to be turned on with respect to time $t-1$ (binary)
$\underline{z}_{ij}^t$	Whether the link (i,j) has to be turned off with respect to time $t-1$ (binary)

Table 5.3: Decision Variables.

t. Two integer variables represent the number of VMs of type l to be turned on (and off) in each SC i at time band t: $\overline{w}_{il}^t$ (and respectively, $\underline{w}_{il}^t$). Binary variable z_{ij}^t is equal to 1 if the path connecting i to j is active at time t, 0 otherwise. Similarly to SCs, $\overline{z}_{ij}^t$ and $\underline{z}_{ij}^t$ indicate whether each path has to be turned on and off with respect to time $t-1$.

$$\sum_{l\in\mathcal{L}}\sum_{j\in\mathcal{N}} x_{ijkl}^t = \lambda_{ik}^t \qquad \forall\, i\in\mathcal{N}, \forall k\in\mathcal{K}, \forall t\in\mathcal{T} \tag{5.1}$$

$$x_{ijkl}^t \le \lambda_{ik}^t m_{kl} \qquad \forall i,j\in\mathcal{N}, \forall k\in\mathcal{K}, \forall l\in\mathcal{L}, \forall t\in\mathcal{T} \tag{5.2}$$

$$w_{il}^t \ge \sum_{j\in\mathcal{N}}\sum_{k\in\mathcal{K}} \frac{D_k x_{jikl}^t}{\overline{U}} \qquad \forall i\in\mathcal{N}, \forall l\in\mathcal{L}, \forall t\in\mathcal{T} \tag{5.3}$$

$$\sum_{l\in\mathcal{L}} P_{il} w_{il}^t \le C_i \qquad \forall i\in\mathcal{N} \tag{5.4}$$

$$\sum_{k\in\mathcal{K}} b_k \sum_{l\in\mathcal{L}} x_{ijkl}^t \le Q_{ij} z_{ij}^t \qquad \forall i,j\in\mathcal{N}, \forall t\in\mathcal{T} \tag{5.5}$$

$$z_{ij}^t = z_{ji}^t \qquad \forall i,j\in\mathcal{N}, \forall t\in\mathcal{T} \tag{5.6}$$

$$\overline{w}_{il}^t \ge w_{il}^t - w_{il}^{t-1} \qquad \forall i\in\mathcal{N}, \forall l\in\mathcal{L}, \forall t\in\mathcal{T} \tag{5.7}$$

$$\underline{w}_{il}^t \ge w_{il}^{t-1} - w_{il}^t \qquad \forall i\in\mathcal{N}, \forall l\in\mathcal{L}, \forall t\in\mathcal{T} \tag{5.8}$$

$$\overline{z}_{il}^t \ge z_{il}^t - z_{il}^{t-1} \qquad \forall i\in\mathcal{N}, \forall l\in\mathcal{L}, \forall t\in\mathcal{T} \tag{5.9}$$

$$\underline{z}_{il}^t \ge z_{il}^{t-1} - z_{il}^t \qquad \forall i\in\mathcal{N}, \forall l\in\mathcal{L}, \forall t\in\mathcal{T} \tag{5.10}$$

Constraints (5.1) make sure that all the incoming traffic is served by any of the SC.

Equations (5.2) ensure that each type of request in each SC is served by the suitable type of VM. Constraints (5.3) determines the number of VMs of type l required to serve the overall incoming workload at site i. As in other literature approaches [179, 173] and currently implemented by Cloud providers (see, e.g., AWS Elastic Beanstalk [11]), these constraints force a suitable number of VMs to be active in i in time slot t, guaranteeing that the average VM utilization is less or equal to a threshold $\overline{U}$. Constraints (5.4) ensure that the number of VMs running at SC i in time t is lower than the available resources.

Constraints (5.5) ensure that traffic on each path does not exceed the available capacity and force z_{ij}^t to be one if the amount of used bandwidth on (i, j) is strictly positive, thus determining the active paths at time t. Constraints (5.6) guarantee that if a path is active in one direction, it is also active in the other, as paths are bidirectional.

Furthermore, VMs and paths switching on and off must be computed along the considered time horizon: (5.7) and (5.8) identify how many VMs have to be switched on or off on with respect to time band $t-1$; equations (5.9) and (5.10) define which paths have to be turned on or switched to idle mode with respect to the previous time band.

SC energy consumption	
α_{il}	energy consumption for running a type l VM in SC i
η_{il}	energy consumption for switching on a type l VM in SC i
θ_{il}	energy consumption for switching off a type l VM in SC i
ρ_i	power usage effectiveness of SC i
Network energy consumption	
γ_{ij}	energy consumption for a router in path (i, j)
δ_{ij}	energy consumption for a router in path (i, j) in idle state
τ_{ij}	energy consumption for switching on a router in path (i, j)
ξ_{ij}	energy consumption for switching off a router in path (i, j)
Costs of brown energy	
c_i^t	cost for energy in SC i at time band t
f_{ij}^t	cost for energy in path (i, j)
Costs and availability of green energy	
Γ_i^t	green energy available in SC i at time band t
g_i^t	cost for green energy in SC i at time band t

Table 5.4: Parameters. Part 2 (Energy related parameters)

We can now introduce the part of the model concerning energy consumption, that is associated to the VMs and routers utilization (see Table 5.4). Parameter α_{il} denotes the energy consumption for running a type l VM in SC i at peak load (in kWh), while η_{il} and θ_{il} are the energy consumption for turning on and off a VM, respectively. The energy consumption, in terms of kWh, for running and keeping idle a single router are denoted with γ_{ij} and δ_{ij}, respectively. Switching on and off a path consumes energy as

well: τ_{ij} and ξ_{ij}. For each unit of energy consumed by the path connecting i to j at time t a cost f_{ij}^t must be paid. In order to take into account service centers cooling energy costs, power distribution and uninterruptible power supply efficiency, we consider the Power Usage Effectiveness (PUE) for SC i, denoted by ρ_i. The PUE is defined as the total service center power divided by the IT equipment power.

Renewable energy sources are considered as well, with cost and availability depending on the sites. We denote with c_i^t and g_i^t the cost for brown and green energy at each SC in the time band t, respectively. Γ_i^t is the renewable energy available at SC i at time t. As for the application workload, green energy sources can be evaluated by relying on prediction techniques [6, 108, 72].

Finally, continuous variable y_i^t models the amount of green energy used in SC i at time t.

We recall that the aim of the problem is to reduce the energy cost along the considered time horizon and to exploit green energy where available. The resulting objective function is:

$$\begin{aligned}\min \quad & \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}}\left\{c_i^t\left[\rho_i\sum_{l\in\mathcal{L}}(\alpha_{il}w_{il}^t+\eta_{il}\overline{w}_{il}^t+\theta_{il}\underline{w}_{il}^t)-y_i^t\right]+g_i^t y_i^t\right\} \\ & +\sum_{t\in\mathcal{T}}\sum_{i,j\in\mathcal{N}} f_{ij}^t R_{ij}\left[\delta_{ij}z_{ij}^t+\tau_{ij}\overline{z}_{ij}^t+\xi_{ij}\underline{z}_{ij}^t+(\gamma_{ij}-\delta_{ij})\frac{\sum_{k\in\mathcal{K}}b_k\sum_{l\in\mathcal{L}}x_{ijkl}^t}{Q_{ij}}\right]\end{aligned} \tag{5.11}$$

where the first term account for SC (VMs) energy consumption and the second term the network consumption. It can be observed that the green energy can be used only to reduce the brown consumption of the VM part. To correctly calculate the value of variable y_i^t additional constraints are needed:

$$y_i^t \le \rho_i\sum_{l\in\mathcal{L}}\left(\alpha_{il}w_{il}^t+\eta_{il}\overline{w}_{il}^t-\theta_{il}\underline{w}_{il}^t\right) \qquad \forall i\in\mathcal{N},\forall t\in\mathcal{T} \tag{5.12}$$

$$y_i^t \le \Gamma_i^t \qquad \forall i\in\mathcal{N},\forall t\in\mathcal{T} \tag{5.13}$$

Equation (5.12) ensures that the green energy used in each SC does not exceed the total energy needed. Equations (5.13) guarantee that the availability of green energy sources in each SC is not exceeded. The lower cost of green energy (we assume $g_i^t < c_i^t$ for all i and t) forces the optimal solution to prefer those SCs which have the possibility to exploit green energy produced in-site, and to use all the green energy available in a site before starting to use brown energy.

Differently from the previous papers on similar topics, our work provided a fully integrated management of service centers and communication network considering variable traffic, energy cost, and green energy availability. We proposed a new MILP model for the integrated problem and we show that it can be solved to optimality in relatively short time (order of minutes) for realistic size instances without the need to rely on heuristic algorithms. Our resource management model has been evaluated under a variety of systems and workload configurations. Nevertheless, for the sake of brevity, we

report here only a case-study to show the potential of our modeling approach to analyze the optimal overall system energy consumption and the impact of green resources on it (for more details on instance generation and computational results, the interested reader can refer to [J13]).

5.1.2 A case-study inspired by Google infrastructure

We perform a cost-benefit evaluation of our solution with respect to a *base scenario* in which the system cannot exploit the network infrastructure to move requests between sites, which is the approach currently used for Cloud resource allocation. Analyses evaluate the costs savings that can be achieved through load redirection, exploiting energy costs variability and green energy availability in multiple locations.

Furthermore, we also evaluate the greenhouse gas emission reduction with respect to a *brown scenario* where no green energy is available. CO_2 emission has been computed through the U.S. Environmental Protection Agency web tool [3].

We consider a case study inspired by the Google infrastructure where we also varied the number of searches to be performed per day. Although Google tends to be quite secretive about its SCs technology, the geographical location of its SCs is known. Google owns 36 SCs spread all over the world [128], in order to have a huge amount of computational resources to satisfy the tens of billions of user requests per day. SCs have been setup near large urban areas close to the end-users in order to reduce network latency. Figure 5.2 and Table 5.5 show details on where SCs are placed.

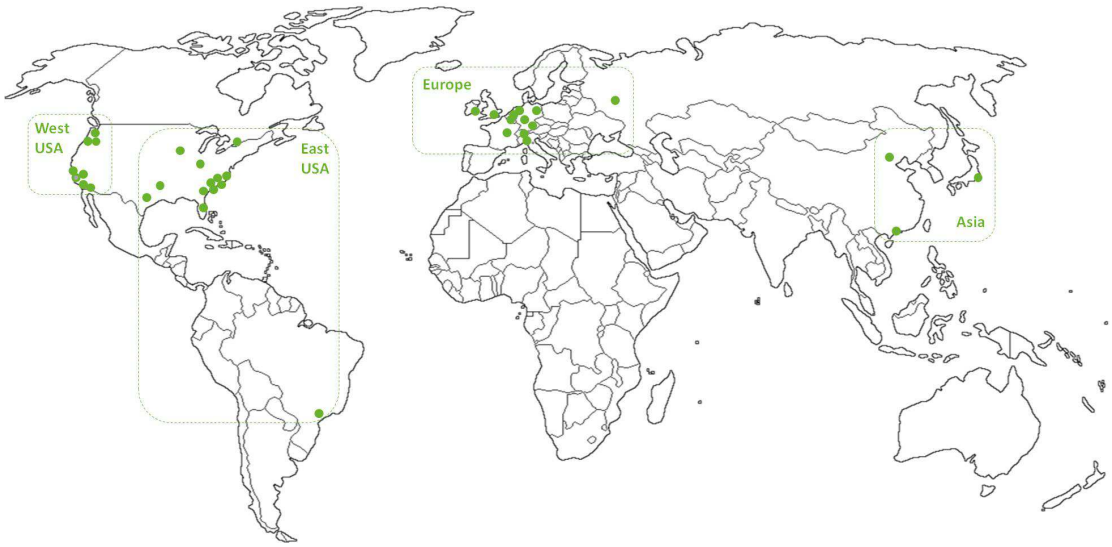


Figure 5.2: Map of Google SCs location considered in our model.

For what concerns SCs capacity, since no specific data is available but only aggregated values have been undisclosed [102], we assume that the servers are almost spread uni-

Service center	*City*	*Country*	*Time zone*
DC1	Mountain View	California (USA)	UTC-08
DC2	Pleasanton	California (USA)	UTC-08
DC3	San Jose	California (USA)	UTC-08
DC4	Los Angeles	California (USA)	UTC-08
DC5	Palo Alto	California (USA)	UTC-08
DC6	Seattle	Washington (USA)	UTC-08
DC7	Portland	Oregon (USA)	UTC-08
DC8	The Dalles	Oregon (USA)	UTC-08
DC9	Chicago	Illinois (USA)	UTC-05
DC10	Atlanta	Georgia (USA)	UTC-05
DC11	Reston	Virginia (USA)	UTC-05
DC12	Ashburn	Virginia (USA)	UTC-05
DC13	Virginia Beach	Virginia (USA)	UTC-05
DC14	Houston	Texas (USA)	UTC-05
DC15	Miami	Florida (USA)	UTC-05
DC16	Lenoir	North Carolina (USA)	UTC-05
DC17	Goose Creek	South Carolina (USA)	UTC-05
DC18	Pryor	Oklahoma (USA)	UTC-05
DC19	Council Bluffs	Iowa (USA)	UTC-05
DC20	Toronto	Canada	UTC-05
DC21	Berlin	Germany	UTC+01
DC22	Frankfurt	Germany	UTC+01
DC23	Munich	Germany	UTC+01
DC24	Zurich	Switzerland	UTC+01
DC25	Groningen	Netherlands	UTC+01
DC26	Mons	Belgium	UTC+01
DC27	Eemshaven	Netherlands	UTC+01
DC28	Paris	France	UTC+01
DC29	London	England	UTC+00
DC30	Dublin	Ireland	UTC+00
DC31	Milan	Italy	UTC+01
DC32	Moscow	Russia	UTC+03
DC33	San Paolo	Brazil	UTC-03
DC34	Tokyo	Japan	UTC+09
DC35	Hong Kong	China	UTC+08
DC36	Beijing	China	UTC+08

Table 5.5: SC locations details.

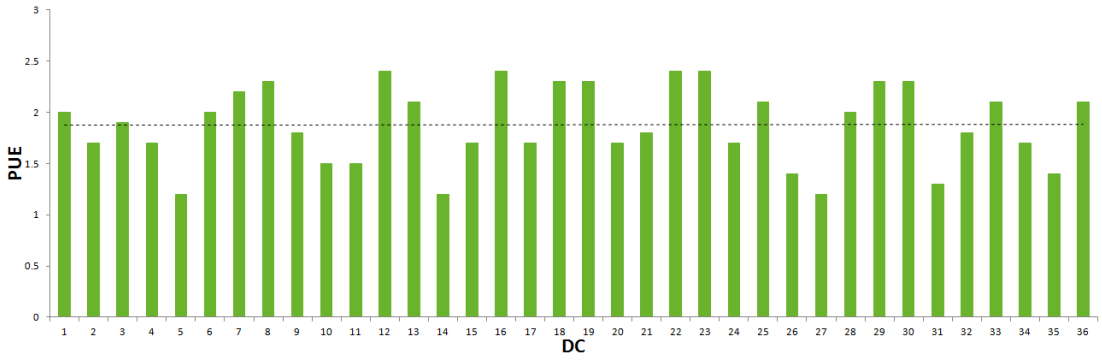


Figure 5.3: Google SC PUE values considered in our analyses.

formly among SCs (a Gaussian random deviation from the mean for each SC is added).

For the PUE value, [129] reports an average value close to 1.8. However, considering SCs geographic location, we can adopt a lower value of PUE for those SCs placed in regions with suitable climatic conditions, like the ones placed in North Europe. In fact, in most moderate climates with temperatures lower than 13°C for 3,000 or more hours per year, new cooling techniques based on free air can eliminate the majority of chillers runtime. For example, the SC in Belgium eliminates chillers, running on free cooling 100% of the time, and reaching in this way a PUE value close to 1.2 [129]. Figure 5.3 shows the PUE value assumed for each SC.

Finally, in order to estimate the total amount of green energy produced by each SC during a single day (parameters Γ_i^t), we assumed that the green energy was proportional to the SC area. The values we used are reported in Figure 5.4.

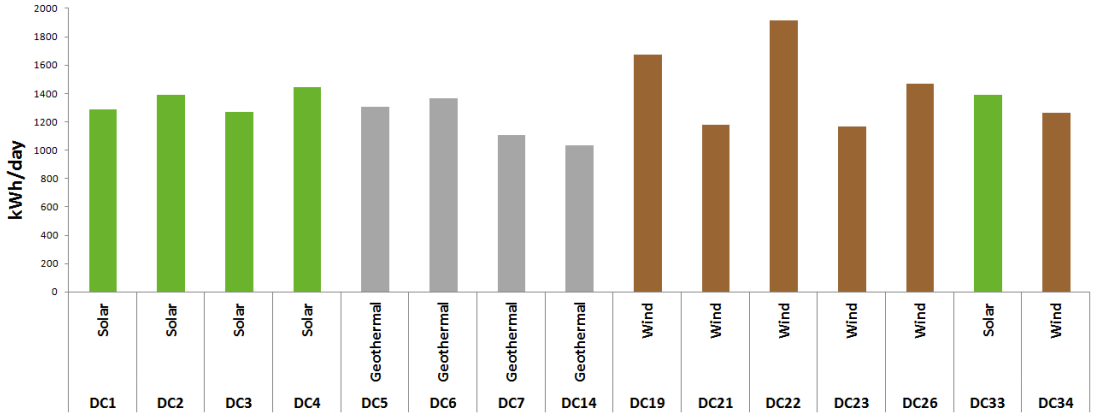


Figure 5.4: Total kWh produced by green Google Service Centers during a single day.

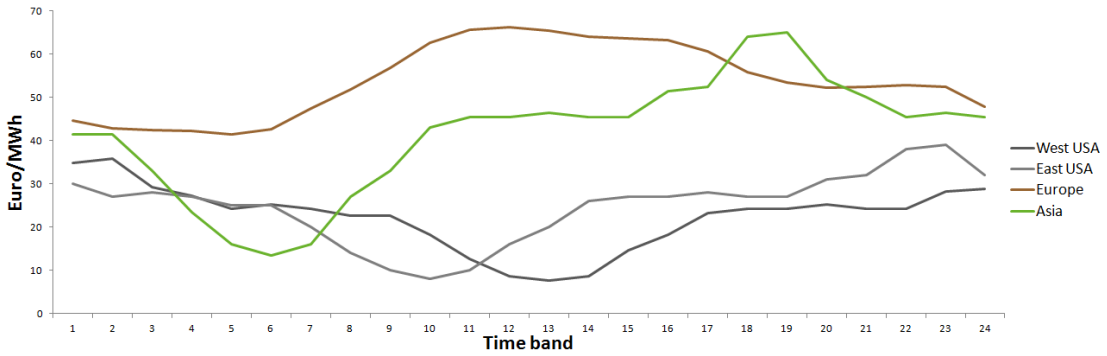


Figure 5.5: Service centers energy cost during each time band (UTC+1).

Analysis

The results we obtained are summarized in figures to allow a better comparison between the solutions that can be obtained by our model (joint DCs management trying to exploit at most green energy resources) and the base scenario (where demands are served locally at the originating DC). Figure 5.6 shows the cost savings that can be achieved by our solution with respect to the base scenario varying the number of searches to be performed per day.

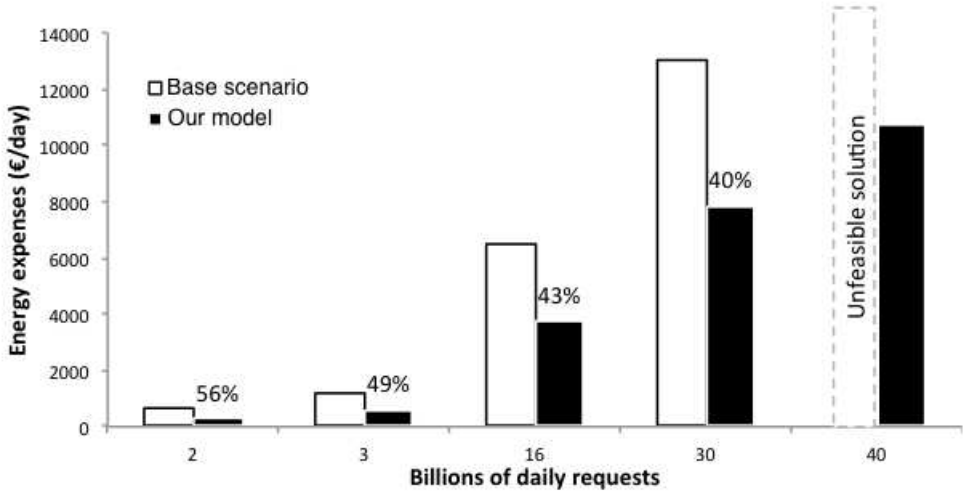
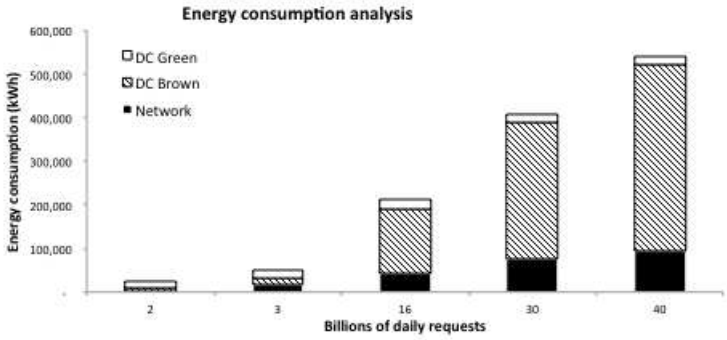


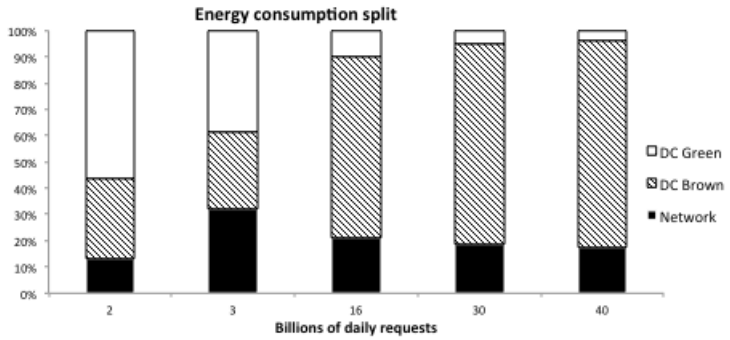
Figure 5.6: Overall cost comparison with respect to the base scenario.

The savings are very significant, ranging between 40% and 56% for the lightest loaded instance. This can be expected, since only a small set of SCs can rely on green energy. Our model forwards as many requests as possible to these sites, until the VMs utilization threshold is reached or the upstream bandwidth is saturated. Additional requests

cannot be served and need to be routed elsewhere, possibly to a non green-enabled SC. Therefore, saving are smaller when the overall workload is higher. This argument is confirmed also by the data reported in Figures 5.7a and 5.7b, which show how the energy consumption is split among the network and the SCs. Whilst the network energy consumption fraction is almost constant and around 15-20%, the test cases with a larger number of daily requests are characterized by a higher usage of brown energy. As it can be expected, with a larger amount of green energy available, a Cloud provider would experience not only savings (due to the lower cost assumed for green energy), but also massive reduction of CO_2 emission, as will be further discussed in the following. It is worth noticing that for the heaviest load instance the base scenario can not find a feasible solution. Hence, the requests redirection allows also to exploit remote SCs available capacity when local resources are saturated.



(a) Absolute energy split.



(b) Percentage energy split .

Figure 5.7: Energy distribution among DCs (both green and brown) and network

It must be observed that, while our approach can achieve lower cost with respect to the base scenario, it may cause a larger energy consumption. As shown in Figure 5.8, this is due to the network transfers and routers crossed along the path between two

sites. However, our model especially fosters the maximum possible amount of "clean" energy usage, unlike the base scenario, which is limited in relying on local energy only. This consideration is confirmed by Figure 5.9, which reports the percentage of green energy used in our model and in the base scenario. The plot clearly shows how our model performs better on all accounts, i.e., it is able to exploit as much green energy as possible by forwarding the requests to the green SCs.

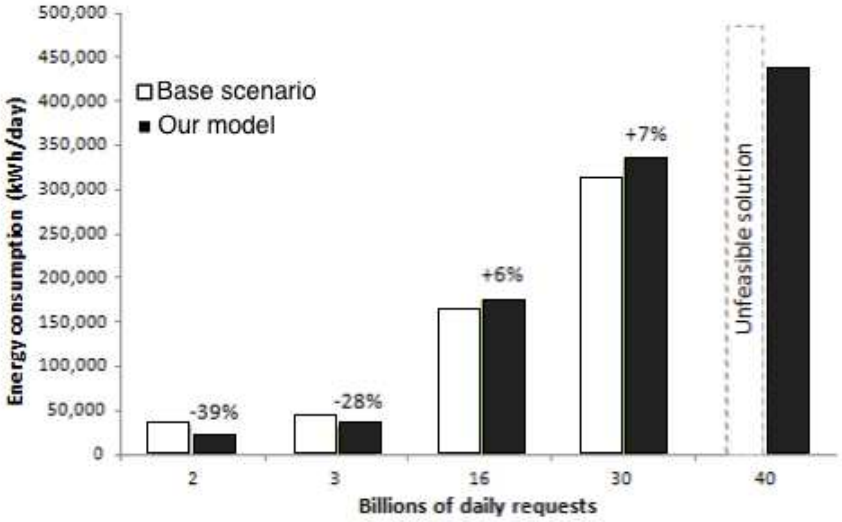


Figure 5.8: Energy consumption comparison with the base scenario.

Furthermore, as a representative example, Figure 5.10 shows the number of active servers for each time band, distinguishing the four macro-regions where SCs are located, while serving 30 billion requests a day. As it could be expected, the trend of each macro-region appears opposite to the energy cost, presented in Figure 5.5: the number of active servers increases when the energy cost decreases, and vice versa. For example, requests are executed in Asia only in the first time bands, where in fact the energy cost has a minimum, while in most of the other time bands the majority of the requests are forwarded to East and West USA, since the energy cost appears always lower with respect to the other areas. However, since also the network costs and the network and SC capacities have to be taken into account, a small amount of requests is served locally also in areas where the energy cost is not optimal.

An interesting comparison can also be made between our model and the brown scenario (see Figure 5.11, where the overall energy consumption is reported). When the workload is light, the energy consumption of our model is higher, while being almost equal in the other cases. It has to be reminded that, for our model, more than 50% of the energy consumed in lightly loaded instances comes from green sources, thus the CO_2 emission

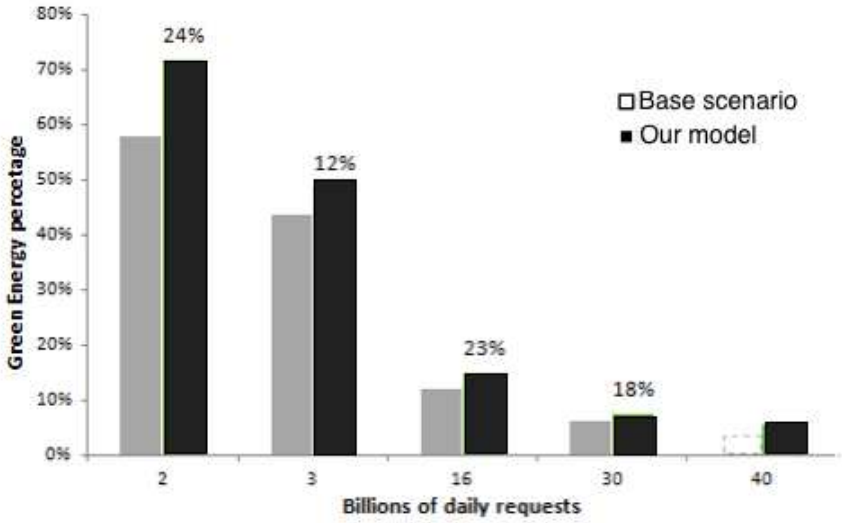


Figure 5.9: Percentage of green energy usage.

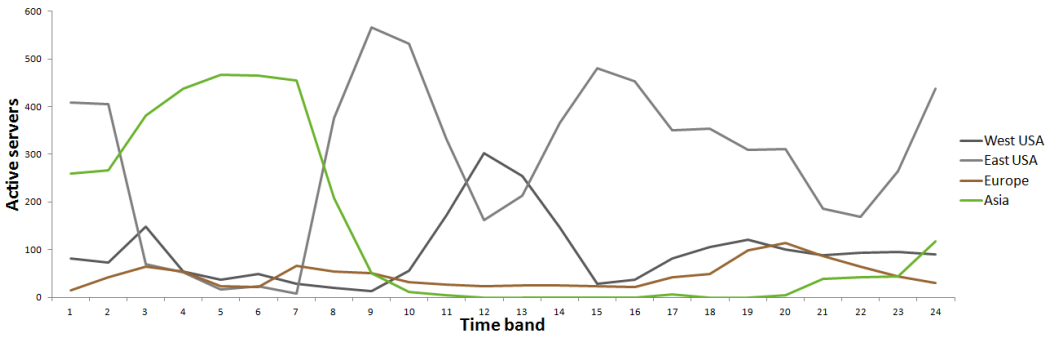


Figure 5.10: Number of active servers for each time band (UTC+1).

results to be lower, as will be discussed in the following. The main explanation for a higher energy consumption is that, considering in particular the 2 billion requests per day scenario, our model consumes more energy for the network, since it has to forward more requests (in percentage) to remote SCs to exploit green energy sources. In the other instances, this problem is marginal, since all the green energy is saturated. Requests are forwarded according to the same criteria as the brown model, thus consuming the same amount of energy.

Finally, Figure 5.12 reports the CO_2 emissions of the brown scenario and of our model. The maximum absolute reduction (around 25 tons of CO_2) occurs in the 40 billion requests per day instance. Our model is able to reduce up to 57% the environment pollution derived from greenhouse gases for the lightest loaded scenario, while this percentage reduces with the workload, advocating a larger adoption of green energy sources for the

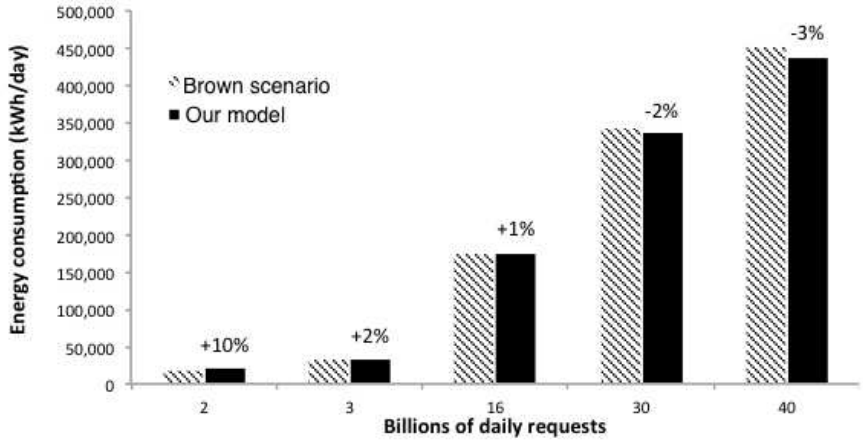


Figure 5.11: Comparison of the energy consumption of our model with the brown scenario.

largest Cloud providers.

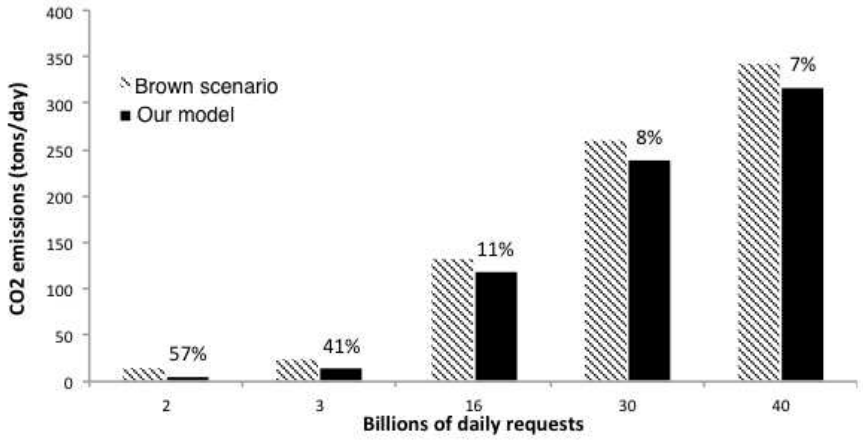


Figure 5.12: CO_2 gas emissions comparison with respect to the brown scenario.

5.2 Virtual Network Functions placement and network routing optimization

The mass diffusion of telecommunication networks and the possibility of accessing email, social networking or cloud computing from mobile devices is exponentially increasing the demand for network services. Network services were up to now provided by proprietary network appliances, such as firewalls, proxies or WAN optimizers, but it has become difficult to integrate and operate such hardware based appliances or to add new functionalities to keep up with the ever increasing demand at a reasonable cost. To cope with this challenge, the Network Functions Virtualization (NFV) paradigm has been recently proposed: hardware based appliances should be replaced by Virtual Network Functions (VNFs) running on generic servers, which will provide the required services in a cheaper and more flexible way.

A key problem arising in implementing the NFV paradigm is the VNF chaining problem: locating VNFs and routing demands so as to guarantee that each demand passes through a sequence (chain) of VNFs that provides the services it needs. The allocation of demands to VNF instances and the demand routing must satisfy quality requirements, such as delay, congestion, minimal resource utilization (CPU, energy, etc). We can schematically describe the VNF chaining problem as follows: we are given a telecommunication network, where the nodes can be connected to computing servers (and/or clouds). Each server is connected to one node, but in general a node can or cannot be connected to a server. Traffic demands must be routed on the network and must be served by a set of VNFs (services), which must be located on computing servers. In some cases, for each demand an order (possibly partial) may be given, according which the VNFs must be traversed. Therefore a traffic demand that needs to access a VNF located in a given server must pass through the node that is directly connected to the computing server itself. Several constraints can be taken into account such as, beside the service order constraint, the VNF capacity constraint, the link capacity constraint, the incompatibility of certain VNFs (thus imposing their installation in different nodes), thus leading to several versions of the problem.

NFV technology has received much attention from both industry and academia, this is attested by an important number of works on NFV addressing challenging aspects spreading from the creation of NFV platforms [152],[94] to the optimization of VNF chaining with respect to, for instance, resource efficiency [115] or competitive goals [C2].

From an optimization point of view, the VNF chaining problem shares features with network design problems (for the demand routing part) and with facility location problems (for the VNF location and the server dimensioning). Both problems are widely studied in the literature, but the combination described by the VNF chaining problem represents a new challenge.

Our contribution to the VNF chaining problem is of two types:

- on the one hand, we proposed mathematical programming models taking into account realistic problem features never considered in the literature before (traffic compression/decompression, different latency profiles, etc.) and performed a de-

tailed analysis of the resulting solutions ([C2],[J2]);

- on the other hand, we investigated the combinatorial structure of the problem (working on a simplified version of it), both in terms of complexity results and problem formulations. This second part of the work was motivated by the presence of different formulations in the ICT literature, which have not been compared.

We believe that a better understanding of the underlying problem structure can be the basis for designing more effective solution methods capable of dealing with the more applicative versions of the problem. Even if the modeling strategies and the results obtained in [C2],[J2] are of a certain interest for the ICT community, for the sake of brevity and "to keep a balance" between application-oriented and methodological contributions, in the following we focus on the study of problem properties and formulations.

Related work

We classify the literature on VNF chaining optimization into two broad categories according to the modeling strategy. One, that is the first proposed in the ICT literature, is based on the Virtual Network Embedding (VNE) paradigm [69]. In the VNE problem a set of virtual networks must be embedded into a physical network: virtual nodes must be mapped on physical ones and virtual links must be mapped on physical paths, while respecting (node and link) capacity constraints. To use such modeling paradigm, each demand is represented by a virtual path-like network, whose origin and destination are known, and intermediate nodes represent the VNF that must be traversed. The second modeling strategy combines routing and location decisions. Even if we believe that the VNE paradigm is not adequate to represent the problem (this point was raised by us in [C2], [J2] and by other authors [9]), and a detailed explanation is reported in Section 5.2.1), we report here some of the most known papers for both modeling perspectives.

In [89], authors introduce a VNE based MIP formulation. In [19] a VNE based ILP formulation is proposed along with a dynamic programming based heuristic to deal with large size instances. Similarly, [127] proposes a VNE based representation of the problem and focuses on the online version of the VNF chaining problem: three greedy algorithms and a tabu-search based heuristic method are proposed to deal with the VNF online mapping and scheduling. The problem of embedding VNF chains is addressed also in [117], where demands may be rejected and the subset of accepted demands must be maximized while limiting the number of service chains considered. In [115], the authors propose an ILP model based on the mapping of VNF chains on a physical network, although without naming VNE explicitly. Precomputed paths for mapping are used. The ILP model is used as a step in a heuristic procedure based on a dichotomic search on the number of located VNFs.

The second modeling perspective, adopted by a handful of papers in the networking literature, exploits the fact that the VNF chaining problem shares features with both facility location and network design problems. Authors in [123] define a model formalizing the VNF chaining problem using a context-free language and propose to use Mixed

Integer Quadratically Constrained Program (MIQCP) for finding the optimal placement of VNFs and chaining them together. In [31], the specific Deep Packet Inspection (DPI) VNF placement problem, where a single type of service is asked, is targeted and modeled as an adaptation of the multicommodity flow problem model. Small instances are solved with a standard ILP solver (GLPK) and for larger instances a centrality-based greedy algorithm is proposed: at each step a new VNF (vDPI) is located in the node that has the highest centrality until all the traffic flows are served or all nodes have a vDPI. In [C2], we provide a MILP formulation accounting for facility location and demand routing with a generic number of services and no fixed order in the chain, taking into account some realistic problem features that were not introduced in the previous models: different latency regimes and traffic compression properties. The work investigates the trade-off between a legacy traffic engineering (TE) goal (namely maximum link utilization) and a combined TE-NFV goal (namely, the sequential optimization of the TE goal and of the VNF installing cost). The work was extended in [J2] where also ordered (or partially ordered) chains are taken into account. Furthermore, a math-heuristic is presented to speed up the solution phase and a numerical comparison with a VNE model approach is proposed. In [9], a similar model is proposed, where additional constraints are added to take into account the incompatibility of certain VNFs and thus imposing that they are located in different nodes. Furthermore, in this work, some demands can be rejected, therefore their routing and VNF assignment can be neglected. A greedy algorithm based on the decomposition of the problem into two steps (routing and location) is proposed. Flows are allocated one by one, and then the VNFs are located on the selected paths.

Although the combined facility location and network design problem has been studied in optimization literature [55], the VNF chaining problem specificities are still not explored in the optimization literature. To the best of our knowledge, we are among the first to investigate in detail the properties of the VNF chaining problem.

5.2.1 VNF chaining problem vs VNE: differences and similarities

This section illustrates the relationship between the VNF chaining problem and the VNE. First of all, we introduce two schematic definitions for the two problems, then we report some general observations that can allow to understand the success of the VNE-based approaches, but also their inherent limits.

We work under the (commonly used) hypothesis that the network link capacity is tighter than the capacity of the connection between computational servers and nodes. Therefore, as each server is connected directly to only one routing node, we use a compact representation where routing nodes and server nodes are collapsed, simply assigning the computational capacity of a server to the associated node. For both problems, let us represent the physical network with a graph $G(N, A)$, where N is the set of (switching/computing) nodes, and A represents the possible directional connections between nodes. Both nodes and arcs are capacitated.

VNF

- A set of (traffic) demands D is given, each demand $k \in D$ is characterized by:

  - a source o_k, a destination t_k, a nominal bandwidth d_k;
  - a sequence of VNFs of different types that must serve the demand (and therefore that must be traversed by it).
- Each VNF istance has
  - a given computational capacity that limits the number of demands that can be assigned to it [1]
  - a capacity occupation (on the node where the VNF will be instantiated) that limits the number of VNF instances that can be installed on each node [2]

VNE

- A set of virtual graphs $\mathcal{D}$ is given $V_k(N_k, A_k)$, where N_k and A_k are the set of virtual nodes and arcs, respectively:
  - both virtual nodes and virtual arcs have a certain demand in term of resources
- Each virtual graph $V_k \in \mathcal{D}$ must be mapped on the physical graph G (Embedding), i.e.
  - each virtual node $n \in V_k$ must be assigned to a physical node $i \in N$ and the capacity of node i must be enough to support the total demand of the virtual nodes assigned to it
  - each virtual arc $(l, m) \in A_k$ must be mapped on a $l-m$-path[3] on G and the capacity of each physical arc $(i, j) \in A$ must be enough to support the total demand generated by the all the virtual arcs mapped on it.

We are aware that these descriptions are quite concise and that several variants of both problems (multiple capacities/resources, multiple instances of the services, etc.) exist, but we believe that are enough for highlight the main differences between the two problems (other probing elements related to more realistic definitions are presented in a recent paper [9]).

In the works that use the VNE paradigm to derive mathematical programming models (and heuristics) for VNF, for each demand k and its associated VNF chain $(VNF_k^1), ..., VNF_k^n$ (where n represent the number of VNFs in the chain), a virtual graph is build such that:

- $V_k = \{o_k, VNF_1^k, ..., VNF_l^k, t_k\}$
- $A_k = \{(o_k, VNF_k^1), (VNF_k^1, VNF_k^2)..., (VNF_k^l, t_k,)\}$
- The virtual node resource request represents the VNFs node occupation capacity.

[1] similarly to facility location problems

[2] similarly to two-levels facility location problems ([1]). VNF instances can be seen as facilities of the lower level and nodes as facilities of the upper level

[3] a path starting from node l and arriving to node m

- The virtual link resource request represents the demand bandwidth.

After the embedding solution is found, the demand routings can be obtained from virtual arcs mappings and the VNF locations from virtual node mapping.

If the order of VNFs in the chain is fixed, the VNF chaining problem in the form we presented (in its general form is not anymore true, see for example [9]) can be considered a special case of VNE where virtual graphs are linear and some virtual nodes have a fixed mapping (origin and destination of the demands must be mapped on the corresponding nodes in the physical graph). Therefore, it is not possible to assess its NP-completeness based on the VNE problem (that is known to be NP-complete in its decisional form). Indeed, we showed ([J3]) that the VNF chaining problem is polynomial in some particular cases and for some network topologies.

When no order (or only a partial one) is imposed for the VNFs, the VNE paradigm is not anymore appropriate, in fact, in this case, a single virtual graph cannot represent all possible VNFs sequences for a single demand. If we want to continue using the VNE-parallel: multiple virtual graphs are needed to representative the chain, but only one of them must be embedded, adding an additional combinatorial layer to the problem.

Nonetheless, the mathematical programming model inspired by VNE appear to be computationally quite effective to solve the VNF chaining problem with fixed order. In short, the reason of the success of VNE-base modeling approaches is related to the fact that in this formulation the routing of each demand is split in subpaths, from origin to the first VNF, from the first VNF to the second, etc. This feature can be exploited leaving aside the parallel with VNE, as it is a well-known strategy of many classical flow problems. Indeed, in our search for formulations, we derived such a strategy without working on the VNE paradigm, and we realized the parallelism just by comparison of formulations.

Therefore, we decided to investigate the relationship between such formulation and the one inspired by network design and facility location problems. To performs such analysis, we started working on a simplified version of the problem. Results will show that this assumption is not so restrictive to make some observations and, more importantly, to give some insight for future research. We believe that, leaving aside the parallelism with the VNE, and taking into account the multicommodity and facility location/binpacking nature of the problem, better formulations can be proposed also for more general versions of the problem (partial order, multiple instances, compression/decompression features of the VNFs, etc).

5.2.2 Problem description, properties and formulations

We focus on a VNF chaining problem where a single type of VNF[4] is considered and any node can be equipped with a single VNF. VNFs and links are capacitated and the

[4]We assume that one VNF type is available, however this assumption is not so restrictive. Indeed, the problem where all the demands ask for the same sequence (same types of VNF and same order) is equivalent to the case with a single VNF type, if VNFs capacity is uniform and a node can host an instance of each VNF type.

number of installed VNF instances is minimized. Further, each demand must be served by an instance of the VNF and must be routed on a simple path, e.g. each demand is not allowed to deviate from its unique path just to "search" the VNF. This assumption is justified by some consideration derived by the real application: first of all, for some telecommunication technologies implementing paths with cycles is not straightforward, for example in label-switching ones where for each packet (of a given demand) labels are used to signal the next node to be reached (therefore, without adding an additional "memory-policy" of the type "when you pass-by a a given node change the forwarding path", paths with cycles cannot be implemented); second, but not less important, the introduction of VNF must be transparent for the user and therefore produce routing similar to the ones used before introducing the function virtualization paradigm (that usually do not have cycles). Let us denote the problem as *Virtual Network Function Placement and Routing with Simple Path* (VNF-PR$_{\text{SP}}$).

Complexity results

We proved that the VNF-PR$_{\text{SP}}$ problem (in its decision form) is NP-complete even if neither nodes nor arcs are capacitated (let call this case UVNF-PR$_{\text{SP}}$) by reduction from the set covering. Instead, the feasibility version of the UVNF-PR$_{\text{SP}}$ is polynomial. It is sufficient to add only one type of capacity (either link or service capacity) to get back to an NP-complete problem even for the feasibility case. The reduction for the link capacity case is from the integer-multicommodity flow and the one for the service capacity case from bin-packing. It is worth to mention that in [116] the problem is proved to be difficult if nodes are capacitated (always reducing from bin-packing). Furthermore, we prove that the UVNF-PR$_{\text{SP}}$ problem is polynomial solvable on some special topologies (complete graphs, rings and tree-like).

Alternative mathematical programming formulations

In this section we present two alternative formulations for the VNF-PR$_{\text{SP}}$ problem.

The first one is directly inspired by network design and facility location problems: a set of variables and constraints represent origin-destination routings and a set of variables and constraints represent the facility location part. Connecting constraints are used to couple the two subproblems. It can be considered as the adapted version of the PR formulation we proposed in [J2] to the VNF-PR$_{\text{SP}}$ problem. The second one, the Split-Path (SP), is based on the decomposition of each demand path into several sub-paths, each associated with a service instance serving the demand. A similar model is presented in [31] (also here a single VNF type is considered). The SP formulation is very close to the VNE-based formulation, nevertheless, it is leaner than it, because there is not need to introduce the concept of virtual graphs, and the mapping of origin/destination nodes. Furthermore, the simple path condition is not guaranteed by the existing VNE-based formulations.

The network is represented by a graph $G(N, A)$, where N represents the set of nodes and A represents the set of capacitated arcs (or links). Let u denote the arc capacity.

An instance of the VNF can be installed on each node in N and can serve a limited amount of demand q. The network demands are represented by the set D: each demand $k \in D$ is characterized by a source (origin) node o_k a destination (terminal) node t_k, a demand amount d_k. The demand must be served (pass through) an instance of the VNF (service)[5], but a demand can pass through a node without using a VNF installed on it[6]. Demands cannot be split and must be routed on simple paths.

The notation is summarized in Table 5.6. As some variables are model dependent, in the last column we report the model in which they are used.

Notation		Model
	Sets	
N	set of nodes	-
A	set of arcs	-
D	set of demands	-
	Capacities (Network and Services)	
u	arc capacity	-
q	service capacity	-
	Demand parameters	
o_k	origin of demand $k \in D$	-
t_k	destination of demand $k \in D$	-
d_k	bandwidth of demand $k \in D$	-
	Variables common to both models (binary)	
y_i	1 if a service is located on node $i \in N$	-
z_i^k	1 if demand $k \in D$ uses the service on node $i \in N$	-
	Routing variables (binary)	
x_{ij}^k	1 if arc $(i,j) \in A$ is used by demand $k \in D$	PR
x_{ij}^{k1}	1 if arc $(i,j) \in A$ is used by demand k on sub-path 1	SP
x_{ij}^{k2}	1 if arc $(i,j) \in A$ is used by demand k on sub-path 2	SP
	TSP-like labeling variables (continuous non-negative)	
π_i^k	position of node $i \in N$ in the path used by demand $k \in D$	PR

Table 5.6: Mathematical notation

In both models, binary variable y_i represents the location of an instance of the VNF on node $i \in N$ and binary variable z_i^k represents the assignment of demand k to the instance of the VNF located on node i. The two models differ in the way the routing is modeled. In both, arc binary variables are used, that are equal to one if a given arc is used by a given demand. In PR, these variables are x_{ij}^k. In SP, the path is explicitly divided into two sub-paths: the first from origin o_k to the service node (described by variables x_{ij}^{k1}) and the second from the service node to the destination t_k (described by

[5]In the following we will use the two terms interchangeably.

[6]In the real application, the demand uses the routing node, but it does not use the server connected to it.

variables x_{ij}^{k2}).

As we want to enlighten the common points and differences between the two models, we present them in parallel, starting from the common part.

$$\min \sum_{i \in N} y_i \tag{5.14}$$

$$\sum_{i \in N} z_i^k = 1 \qquad \forall k \in D \tag{5.15}$$

$$z_i^k \leq y_i \qquad \forall k \in D, i \in N \tag{5.16}$$

$$\sum_{k \in D} d_k z_i^k \leq q \qquad \forall i \in N \tag{5.17}$$

The objective function (5.14) minimizes the sum of the opened services (i.e., instances of the VNF). Constraints (5.15) impose that each demand is assigned to exactly one instance of the service. Inequalities (5.16) guarantee that if no VNF instance is installed on a node, then no demand can be assigned to it. Constraints (5.17) impose that each instance of the VNF can serve a maximum quantity q of demand.

The link capacity constraints are similar for the two models:

SP:

$$\sum_{k \in D} d_k (x_{ij}^{k1} + x_{ij}^{k2}) \leq u \qquad \forall (i,j) \in A \tag{5.18}$$

PR:

$$\sum_{k \in D} d_k x_{ij}^{k} \leq u \qquad \forall (i,j) \in A \tag{5.19}$$

We now present the constraints characterizing each formulation. The main difference is in the way the routing is managed, and, as a consequence, in how the models deal with the coherence between service assignment and routing. In short, in the SP formulation routing and assignment coherence is implied by the (modified) flow balance constraints, while in the PR formulation the routing is implied by the classical flow balance constraints and the consistency between assignment and routing must be enforced adding using additional constraints.

SP:

$$\sum_{j:(i,j) \in A} x_{ij}^{k1} - \sum_{j:(j,i) \in A} x_{ji}^{k1} = \begin{cases} 1 - z_i^k & \text{if } i = o_k \\ -z_i^k & \text{otherwise} \end{cases} \quad \forall k \in D,\ i \in N \tag{5.20}$$

$$\sum_{j:(i,j) \in A} x_{ij}^{k2} - \sum_{j:(j,i) \in A} x_{ji}^{k2} = \begin{cases} z_i^k - 1 & \text{if } i = t_k \\ z_i^k & \text{otherwise} \end{cases} \quad \forall k \in D,\ i \in N \tag{5.21}$$

$$\sum_{j:(j,i)\in A} (x_{ji}^{k1} + x_{ji}^{k2}) \leq 1 \; \forall k \in D, i \in N \tag{5.22}$$

$$\sum_{j:(i,j)\in A} (x_{ij}^{k1} + x_{ij}^{k2}) \leq 1 \; \forall k \in D, i \in N \tag{5.23}$$

Each demand is routed on two sub-paths: from the source node to the VNF node (equations (5.20)), then from the VNF node to the destination node (equations (5.21)). These two constraints impose that the routing of the demand passes through the VNF instance assigned to the demand itself, therefore ensure that the assignment is consistent.

Simple path routing[7] is imposed by constraints (5.22) and (5.23): each demand can pass through a node at most once.

PR:

$$\sum_{j:(i,j)\in A} x_{ij}^k - \sum_{j:(j,i)\in A} x_{ji}^k = \begin{cases} 1 \text{ if } i = o_k \\ -1 \text{ if } i = t_k \\ 0 \text{ otherwise} \end{cases} \quad \forall k \in D, \; i \in N \tag{5.24}$$

$$z_i^k \leq \sum_{(j,i)\in A} x_{ji}^k \; \forall k \in D, \; i \in N \setminus \{o_k\} \tag{5.25}$$

$$\pi_j^k \geq \pi_i^k + x_{ij}^k - |N| \; (1 - x_{ij}^k) \; \forall k \in D, \; (i,j) \in A \tag{5.26}$$

Each demand is routed from its source to its destination with the classical flow balance constraints (5.24) and it is forced to pass through a VNF instance by constraints (5.25), that impose that a demand k can be assigned to a service located on node i only if the routing path of the demand passes through the given node. The simple path and the elimination of isolated cycles are enforced using the TSP-like labeling variables π and constraints (5.26): continuous variables π_i^k represents the position of node i in the routing path of demand k. We can observe that without constraints (5.26) isolated cycles hosting a service can appear (even in integer solutions).

Both models can be generalized to take into account multiple service types[8] with a fixed order. However, when a partial (or any) order is asked, the SP formulation cannot

[7] Isolated cycles with no service can be part of a feasible solution. Such cycles can be removed obtaining a cycle-free equivalent feasible solution.

[8] in the SP model for n services will result in $n-1$ sub-paths

be straightforwardly extended. This is due to the fact that the routing part and the location part are directly coupled by the notion of sub-path, and therefore, to have a VNF chain with a free order, new variables are needed to decouple them and generalize the model. On the other hand, such coupling represent the strength of this formulation.

Instead, the PR model can be simply generalized to deal with any imposed order (full, partial, none). In fact, variables π can be used, together with some additional constraints, to impose a given full or partial order of the services along the demand path (see for example [C2]).

Relation between the two formulations

We proved that the SP formulation produces a continuous relaxation bound that is always not worse than the one produced by PR. Furthermore, we produced evidence that the gap between the two relaxations is greater than zero on some instances.

We will not report here the full proofs, however it is worth to mention the general idea of the proof, in fact this can give some insights on the structure of the feasible regions of the two relaxations (SP_r and PR_r).

Remark 1 *The SP_r forbids demands to be served, even partially, by a VNF instance installed on an isolated cycle. Instead PR accepts solutions with an isolated cycle and a partial service installed on it* [9].

Therefore we can partition the feasible region of the SP_r into two subsets: in the first one no isolated cycles exist while in the second one isolated cycles are present, but they do not host a service. Any solution of the second subset has an equivalent in the first one (as in classical flow problems).

The last step is to prove that any solution belonging to the first subset can be transformed into a solution of PR_r. The mapping of x,y and z is quite straightforward, therefore here we discuss only the mapping of variables π (the complete proof can be found in [W1]).

For any demand k, the flow in a feasible continuous solution can be divided into flow on simple paths and flow on cycles. Let us denote with P_k the set of simple paths and with C_k the set of cycles. We can distinguish between two type of cycles (see Figure 5.13): cycles sharing some nodes with a simple path (Figure 5.13a) and isolated cycles (Figure 5.13b).

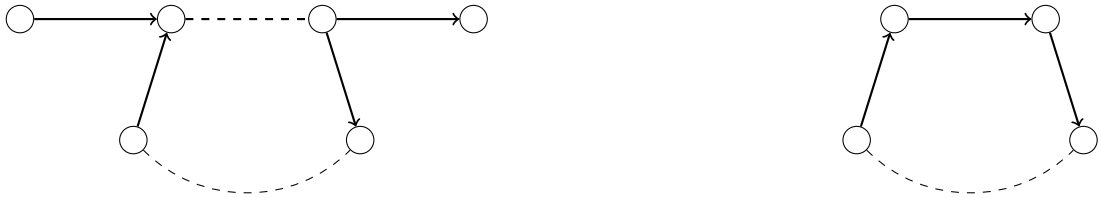
Let us consider a demand k and its routing. We can build the graph induced by the simple paths P_k:

$$G^k(N_k, A_k)$$

where

- an arc $(i,j) \in A$ belongs also to A_k if $x_{ij}^{k1} + x_{ij}^{k2} > 0$ and $\exists p \in P_k : (i,j) \in p$, i.e. the arc belongs at least to a simple path;

[9] as for TSP, isolated loop elimination constraints (5.26) are effective only in the integer formulation.



(a) Cycle sharing nodes with a simple path

(b) Isolated cycle

Figure 5.13: Examples of cycle sharing nodes and of isolated cycle

- a node $\hat{\imath} \in N$ belongs also to N_k if there exists an arc ($(\hat{\imath}, j)$ or $(j, \hat{\imath})$) in A_k.

For any arc $(i, j) \in G^k$, we define the following cost $c_{ij} = x_{ij}^{k1} + x_{ij}^{k2} = x_{ij}^k$.

G^k is acyclic, as it does not contain arcs that belong only to cycles (C^k). Thus, we can define π_j^k as the longest path from node o_k to node j:

- $\pi_{o_k}^k = 0$
- $\pi_j^k = \max_{i:(i,\hat{\jmath}) \in A^k} \{\pi_i^k + x_{ij}^k\}$

Such values satisfy constraints (5.26) by construction for all couples of nodes both belonging to the path.

We now need to determine a value of π for the remaining nodes (nodes belonging only to a cycle[10], as they are not present in $G^k(N_k, A_k)$) in such a way that every couple of consecutive nodes corresponding to an arc belonging only to the cycle (nodes from m to l in Figure 5.14) satisfies constraints (5.26).

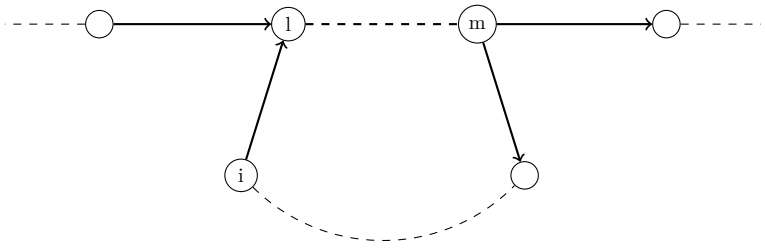


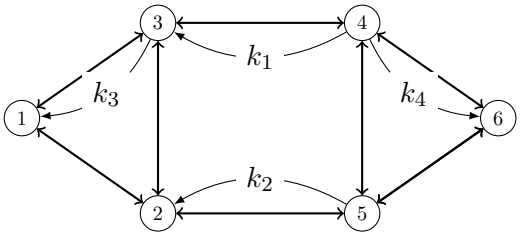
Figure 5.14: An example of a cycle sharing some nodes with a simple path

Let us consider the two nodes l and m, corresponding to the smallest (π_l^k) and largest (π_m^k) value of variables π on the cycle N_C, respectively. If for all the nodes $i \in N_c$, we chose:

$$\pi_i^k = \frac{\pi_m^k + \pi_l^k}{2}$$

[10]for nodes that do not belong to cycles or paths, any value of π can be assigned because the corresponding routing variables are zero.

constraints (5.26) are always satisfied. The proof is based on the fact that variables π are bounded by $|N|-1$ and that the (SP_r) routing variables for any arc (i,j) belonging to the cycle, but not a path, are bounded by $\frac{1}{2}$.



(a) Topology and demands distribution.

k	$o_k \to t_k$	d_k
1	$4 \to 3$	**5**
2	$5 \to 2$	**5**
3	$3 \to 1$	$\forall d_3 \in (0,5]$
4	$4 \to 6$	$\forall d_4 \in (0,5]$

(b) Demands.

Figure 5.15: The SP relaxation bound is strictly better than the one of PR: a numerical example

We now provide an instance in which the bound provided by SP is stricter than the one provided by PR. Let us consider the symmetric graph in Figure 5.15a (in the following figures, each pair of symmetric arcs is represented by the corresponding edge). The services are uncapacitated, and the link capacity is 5. There are 4 demands, whose characteristics are listed in Table 5.15b. The continuous relaxation provided by SP is equal to 2, while the continuous relaxation provided by PR is 1, for any value of the demands k_3 and k_4 in $(0,5]$.

demand	paths	f_p	node (i)	z_i^k
k_1	$p^1 : \mathbf{4} \to 5 \to 2 \to 3$	1	4	1
k_2	$p^2 : 5 \to \mathbf{4} \to 3 \to 2$	1	4	1
k_3	$p^3 : \mathbf{3} \to 1$	1	3	1
k_4	$p^4 : \mathbf{4} \to 6$	1	4	1
Installed services				
$y_3 = 1 \quad y_4 = 1$				

Table 5.7: Routing and assignment/location in the continuous relaxation of SP

We report an optimal solution of SP_r formulation in Table 5.7 and a solution of the $PR-r$ one in Table 5.8: the assignment of service to nodes is reported in bold. As for the SP formulation, one service is located on node 4 and one on node 3. Demands k_1, k_2 and k_4 are served by the service located on node 4, while demand k_3 is served by the service located on node 3. As for the solution of PR, half service is installed on node 4 and the other half on node 5; each demand uses both services.

This example allows us to make some observations on the structure of the feasible space of the two relaxations, and show some general properties. Indeed, we proved

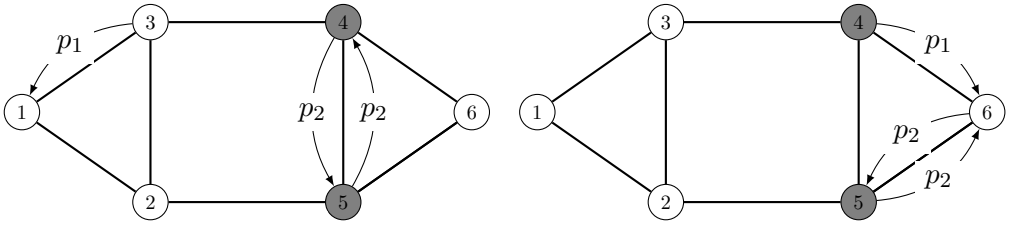
demand	paths	f_p	node (i)	z_i^k
k_1	$p_1 : \mathbf{4} \to 3$	0.5	4	0.5
	$p_2 : 4 \to \mathbf{5} \to 2 \to 3$	0.5	5	0.5
k_2	$p_1 : \mathbf{5} \to 2$	0.5	5	0.5
	$p_2 : 5 \to \mathbf{4} \to 3 \to 2$	0.5	4	0.5
k_3	$p_1 : 3 \to 1$	1	-	
	$p_2 : 5 \to \mathbf{4} \to \mathbf{5}$	0.5	4	0.5
			5	0.5
k_4	$p_1 : \mathbf{4} \to 6$	1	4	0.5
	$p_2 : 5 \to 6 \to \mathbf{5}$	0.5	5	0.5
Installed services				
$y_4 = 0.5 \quad y_5 = 0.5$				

Table 5.8: Routing and assignment/location solution in the continuous relaxation of PR

(see [W1]) that some family of solutions are feasible for PR_r, but not for SP_r. In this example, we can see two of them (see Figure 5.16):

- any solution where a source-destination path does not pass through any service node and an isolated loop hosts a (partial) service (as for demand k_3) (see Remark 1)
- any solution where the routing variables (f_p in the table) have a different value from the assignment variables (path p_2 for demand k_3 and path p_1 for demand k_4).

Indeed, for the SP_r formulation, routing and assignment variables must be consistent. In this sense, the SP model shares some similarities with the min-cost-flow problem: service nodes can be imagined as both sink and source of a demand and assignment variables z_i^k as the quantity of demand that is absorbed/produced by the node (depending if we are looking at the sub-path entering or exiting the node). The main difference is that in our problem the sink/source nodes are decision variables and not given data.



(a) Routing of demand k_3 (b) Routing of demand k_4

Figure 5.16: Routing solution for demand k_3 and k_4 for the PR_r

5.2.3 A numerical comparison

We tested the two formulations on 16 networks (with minimum 10 nodes and maximum 28 nodes) from the SNDLib ([141]). The topologies and the traffic demands (i.e., source, destination, amount of flow) are taken directly from SNDLib. Different capacity values were generated to analyze the impact of the VNF and link capacity:

- As for the links, two levels of capacity have been generated: low and high. The high capacity is such that all the demands can be routed on a single link, thus leading to uncapacitated link instances. The low capacity is computed as the minimum capacity such that a feasible routing exists, neglecting the services.
- As for the services, three levels of capacity are considered: low, medium and high. The high capacity is computed so as to guarantee that all the demands can be served by a single VNF (service uncapacitated instances). Low VNF capacity is twice the total amount of the demands divided by the number of nodes, that is, we need to install a VNF in at least half of the nodes (for lower values many instances were not feasible due to network topologies). Medium capacity is the average between high and low.

By combination of these capacity cases, we obtained 96 instances. In the following, we denote with *h, m* and *l* the *high, medium and low* capacity level respectively. Therefore, for example instances marked with *l_h* are the ones where service capacity assumes the lowest value and the link the higher (therefore uncapacitated with respect of links). Data are summarized in Table 5.9: columns two to four report the network features from SNDLib (number of nodes, number of links and number of demands). Column five gives the sum of the demand amount based on the SNDLib values. Such value correspond to the high capacity value both for links and services. The last three columns give the capacity values both for services and links.

Models are implemented in AMPL and solved with IBM ILOG CPLEX 12.7.1.0 on an Intel Xeon, CPU E5-1620 v2 (4 cores), 3.7 GHz with 32 GB of RAM with a time limit of 3600s and a tree memory limit of 3000 MB.

Enriching the formulation

We try to improve the formulation by adding different families of valid inequalities, we report results only for the two most effective ones. To derive the first one, we can observe that the total demand that can be served by a service located on the node i is limited not only by the service capacity, but also by the overall capacity of incident links (except for the demands served in the origin node i). The same reasoning can be applied to the outgoing links. Therefore, the maximal demand that can be served by a node can be calculated as follows:

$$\bar{q_i} = \min \left\{ q, \max \left\{ \sum_{(i,j)\in A} u + \sum_{k\in D: t_k = i} d_k, \sum_{(j,i)\in A} u + \sum_{k\in D: o_k = i} d_k \right\} \right\} \tag{5.27}$$

Data from SNDLib					Capacity		
Network	$\|N\|$	$\|L\|$	$\|D\|$	$\sum_{k\in D} d_k$ (high cap)	Service medium	Service low	Link low
abilene	12	15	132	3000002	1750001	500000	829282
atlanta	15	22	210	136726	77478	18230	19404
dfn-bwin	10	45	90	548388	329032	109677	55916
di-yuan	11	42	22	53	31	9	5
france	25	45	300	99830	53908	7986	9413
geant	22	36	462	2999992	1636359	272726	359868
janos-us	26	42	650	80000	43076	6153	7624
newyork	16	49	240	1774	997	221	66
nobel-eu	28	41	378	1898	1016	135	214
nobel-germany	17	26	121	660	368	77	74
nobel-us	14	21	91	5420	3097	774	486
norway	27	51	702	5348	2872	396	358
pdh	11	34	24	4621	2730	840	384
polska	12	18	66	9943	5800	1657	995
sun	27	51	67	476	255	35	53
ta1	24	51	396	10127249	5485593	843937	819678

Table 5.9: Instance details

This allow to obtain a strengthened version of constraint (5.17) (using both the service location variable y_i and the capacity $\bar{q}_i$):

$$\textbf{VI1:} \quad \sum_{k\in D} d_k \; z_i^k \;\leq\; \bar{q}_i \; y_i \; \forall i \in N \tag{5.28}$$

When the capacity of a single VNF instance is not large enough to serve all the demands, a bound on the number of needed VNF instances can be calculated (similarly to bin-packing problems [11]):

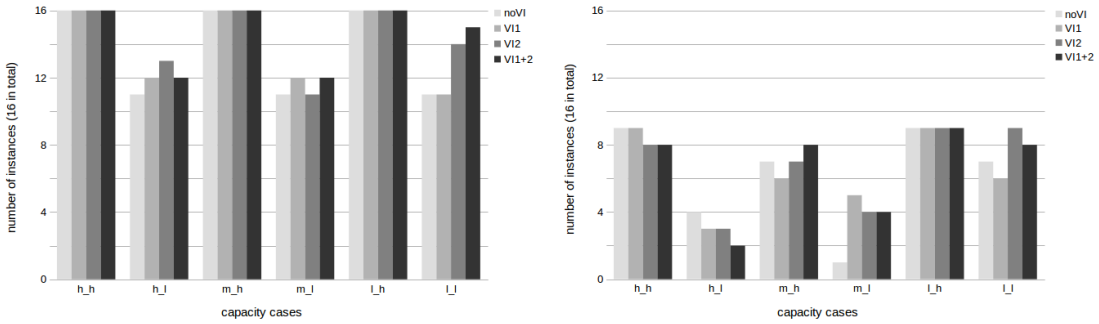
$$\textbf{VI2:} \quad \sum_{i\in N} y_i \;\geq\; \left\lceil \frac{\sum_{k\in D} d_k}{q} \right\rceil \; \forall i \in N \tag{5.29}$$

ILP results

In Figures 5.17a- 5.17b, the number of optimal solutions found by the SP and respectively PR are reported.

SP finds always the optimal solution when the link capacity is high, whereas PR finds the optimal solution in just a around half of the instances. When the link capacity

[11] Indeed, we are currently testing the use of the bound obtained solving the underlying bin-packing problem associated to VNF placement



(a) Number of optimal solutions SP.

(b) Number of optimal solutions PR.

Figure 5.17: Comparison between the number of optimal solutions found by SP and PR formulations

is low, SP cannot find the optimal solution for five instances out of 16 when no VIs are added, but this number improves adding VIs. Furthermore, when SP cannot prove optimality, the gap is still reasonable (around 27% for janos_us instance with h_l and m_l capacities). On the contrary, PR can solve to optimality less than half of the instances (for the m_l case less than a quarter). Adding VIs not always improves the performance, for example in the h_l case the number of optimal solutions is reduced. In general, the impact of $VI1$ is more evident on instances where the link capacity is low and the service capacity is high. This is reasonable, because on these instances the multicommodity nature of the problem emerges and the capacity of incident links is the one that limits the access to services. Symmetrically, when the service capacity is low, the VI2 shows its effectiveness more clearly.

In Table 5.10, computational times (in seconds) are reported. The average computational time is calculated also including the cases where the time limit (3600s) is reached. In this way, the impact of not finding a solution is taken into account. Even in terms of computational time, the SP formulation performs better than the PR formulation. Adding both valid inequalities seems to produce the best results, reducing in almost all cases the total computational time.

We are currently extending PR and SP formulations to the multi-service and multi-sequence case to continue our analysis on more realistic instances. Furthermore, taking into account the theoretical results we derived (bin-packing/multicommodity nature, presence of cycles hosting a service for the PR formulation, etc), we are studying exact and heuristic algorithm to solve more effectively the problem.

5.3 Conclusions

In this chapter we presented two contributions related to problems arising from the convergence of network and computing systems management. This is a quite new per-

cap	SP				PR			
	noVI	VI1	VI2	VI1+2	noVI	VI1	VI2	VI1+2
h_h	**2.46**	2.54	6.46	6.79	**1679.58**	1681.81	1919.80	1928.85
h_l	1528.01	1427.23	**1026.41**	1294.09	**2936.61**	3198.52	3087.09	3034.33
m_h	351.03	63.77	53.10	**10.53**	2301.89	2212.81	1812.10	**1670.79**
m_l	1605.13	1396.10	1243.67	**1228.73**	3388.13	**2705.17**	2754.96	2942.03
l_h	120.04	53.77	7.15	**5.16**	1474.12	**1167.35**	1581.80	1588.63
l_l	1420.11	1569.64	445.12	**369.82**	1864.82	1704.02	**1665.86**	1703.43

Table 5.10: Average computational times (in seconds) taking into account also the instances in timelimit (3600s)

spective for the ICT community, where historically the telecommunication and the cloud community were separated. The new optimization problems resulting from this convergence and the challenge related to their solution are one of the two axis around which my research project is developed.

The first contribution is on the joint management of service centers and communication networks. A mathematical programming model is proposed to analyze the impact of managing jointly multiple services centers, allowing to use the network to redirect demands originated in one service center to another one. The possibility to move "smartly" demands allows to fully exploit green-energy sources (as solar, wind or geothermic), that for they nature are volatile, and therefore must be used when (and where) available. Our proposed model and the analysis of the results we obtained show clearly the advantages of such a joint management advocating a larger adoption of green energy sources for the largest Cloud provides.

This application can be considered as the "trait d'union" between my previous works on network design and management (with some contributions also on data-centers management) and the most recent works on the Virtual Network Functions (VNF) placement and routing, that is the second contribution presented in this chapter. The resulting optimization problem is a novel combination of a network design problem and a facility location one. Our contribution to this problem is twofold: first, we provide mathematical programming models and analysis on realistic problem setting (this contribution is just mentioned), second, but not less important, we provide an analysis of the properties of the problem in terms of computational complexity and we compare, theoretically and numerically, the current state of the art formulations.

6 Perspectives: toward multi-scale and heuristic decomposition methods for complex systems optimization

I plan to pursue my research along two lines: I will continue to work on discrete optimization methods for ICT problems and I will reinforce my research investment on optimization methods for process synthesis. Following this second line, starting from non-linear non-convex optimization, where most of my non-linear optimization expertise lies, I want to wider my knowledge on non-convex MINLP, that combines the challenges of discrete and non-linear optimization together.

6.1 Discrete optimization problems arising in ICT

ICT is going through a significant change, due to massification of services. Key enablers of this change are the virtualization of telecommunication networks and the convergence between telecommunication networks and computing systems. This change is very interesting from the ICT perspective, as it promises to allow managing a wide variety of personalized services for an ever increasing number of potential clients in an effective way in terms of monetary and, most important, environmental cost. Nevertheless, it poses an important challenge: how to manage effectively a system that becomes more and more complex and that integrates so different aspects? How to take into account the dynamical aspects due to the frequent changes of the system in terms of available resources? How dynamically serve the ever changing and increasing demand load?

Such challenge is an opportunity for the optimization community: in fact, moving to a virtualized environment allows to have (at least partial) centralized control, furthermore the abstraction introduced by virtualization allows more freedom in the management. Such features allow to design and effectively use centralized optimization methods based on demand predictions. On the other hand, as the resulting systems are complex, huge in size, and dynamically changing, the state of the art optimization tools are not enough to efficiently tackle the resulting optimization problems. Indeed, solutions need to be economical viable for all the involved economical actors, usable from a client perspective and last, but not least, environmental sustainable. All of it means that the designed approaches need to be enough accurate to capture the essential nature of the problems, and at the same time fast enough to be applicable in a real context.

Key aspects to successfully tackle the management of the new generation ICT systems are the capability to obtain quite accurate (service and resources) demand predictions and the development of hierarchical models that can allow to describe the system with

different levels of granularity.

A central point is how to combine these key aspects to allow efficiently solving the original problem through a sequence of smallest scale problems. To give a practical example, in [J16], we proposed a hierarchical framework to manage very large cloud platforms. At the highest level of the hierarchy, a Central Manager (CM) using one-day ahead demand predictions aggregates demands with the aim of obtaining clusters with a "homogeneous" profile. Furthermore, the CM partitions the available pool of servers. In this way, on an hourly basis and with updated demand prediction, each Application Manager can determine for a single cluster of demands and a given subset of servers: the allocation of demands to servers, the VM load balancing and capacity allocation and the servers switch on/off. Finally, on a short-term base (i.e. 5 minutes), load balancing, capacity allocation and frequency scaling are repeated to adjust the system to the "real" demand. The hierarchical model has different advantages: allows to update predictions at each level of granularity, therefore increasing their accuracy and allowing to adjust the current solution; it allows to solve subproblems of smaller scale on "homogeneous" clusters, therefore balancing at least in a "raw" way the global charge of the system. A similar paradigm can be extended to other ICT problems, as they share features with the previous example: a demand that arrives along the time with a periodical or pseudo periodical nature, a fluctuation of the demand along the day that allows to aggregate demands to reduce the system utilization, a hierarchical nature of the system management.

A central point for the success of hierarchical approaches is to solve efficiently the different granularity subproblems, in fact, the quality of the solution of these problem can impact on the overall system performances. Even if smaller in size than the original problem, they can be very challenging, in fact, from an optimization point of view many of the new ICT ecosystems cannot be traced back to existing optimization problems, but they can be viewed as an original combination of different optimization problems: scheduling, network design, facility location. I give here some illustrative examples.

- The problem of managing decentralized data-centers can be view as the combination of an allocation/scheduling problem and network design one. We considered a high granularity model (see Section 5.1.1), allowing to determine a high level strategy for traffic migration between data-centers considering the impact of the network. This model can be integrated with lower granularity models to manage in detail, from the one hand, the migration of VMs at the data-center level, and from the other hand, the network impact (capacities, consumption, delay).

- The VNF chaining problem can be modeled as the combination of a facility location problem and a network routing and/or loading/design one (see Chapter 5.2).

- As a last example, it is worth to mention the server consolidation problem in a single data-center. In the past, the impact of the network communication was taken into account with very approximated models, i.e. imposing penalties on VM migration. Nowadays, as network performances are not increasing as fast as the computational ones, and the size of data-centers is always increasing (due to the

"big-data revolution"), this approach is not viable anymore, and the impact of the network must be considered explicitly (see for example, the recent paper [133]). Thus, scheduling/assignment problems merge with network design ones.

The panorama of research perspective is quite huge and some of the research leads can be fruitful followed only in a perspective of large collaborative projects, putting together researches from different domains. My contribution to the field will focus on the study of the new optimization problems arising from the aforementioned convergences between communication networks and computing systems and the willingness to bring not only new optimization approaches, but also a, possibly, "far away look" from the application perspective allowing to show similarity with existing optimization problems and solutions (as we did in [J7] for the green-networking case, see Section 4.1).

More concretely, in the short period, I will focus on the VNFs placement and routing problem (see Chapter 5.2). We worked on a simplified version of the problem where demands ask only for a single service and the objective is to minimize the number of installed service instances. Starting from here, I want to pursuit the study along two lines: one is to develop exact and heuristic strategies to solve larger size instances, the second is to consider additional features of the problem (routing costs, multiple services, compression/decompression etc). These two lines must converge to propose solutions for real application cases.

Concerning the first line, a research lead is to exploit the combinatorial nature of the problem, such as the bin-packing substructure related to the demand/service assignment. We observed that valid inequalities based on it are quite effective in improving the overall formulation performances. The general idea is to solve a bin-packing problem to aggregate demands in clusters and solve the problem using them instead of independent demands. This strategy not only allows to reduce the problem size (the number of demands has a strong impact on the computational times), but also can produce, if well-exploited, a guaranteed optimality gap (as the bin-packing solution is a valid lower bound for the problem). The bin-packing must be enriched with specialized cuts to enforce feasibility, for example max-flow based cuts for taking into account link capacities. Following this idea, exact methods based on cut generation will be developed. To tackle large size instances, the procedure can be combined with a local search procedure to change the initial clusters and/or to increase their number. Here again, if the addition of cluster is limited, the resulting solution can have an optimality gap guarantee. Concerning the additional features, many of them were already considered in our preliminary work ([C2],[J2]), but our main contribution was oriented to show the impact of such features on real case studies and to develop efficient optimization methods. Therefore, my research will concentrate on extending the formulations we proposed for these cases and study their properties and computational performances.

Ortogonal to the aspects I have already illustrated, there is another open challenge: the reduction of the energy consumption and CO_2 emissions. Even if for the ICT community the green perspective seems resolved or outdated, I believe that it is still an actual problem. It is enough to look at the European research agenda to see it. From an optimization point of view, taking into account energy consumption can change the

structure of the addressed problem. In fact, the green perspective asks for the system to adapt to the demand in a dynamical way, introducing two levels in the problem: one related to the physical structure (device installed) and the other on the dynamical modulation of the devices according to an energy consumption reduction (devices switch on/off). This can allow to exploit this substructure to efficiently tackle the problem with decomposition methods or using projected formulations. Furthermore, if we consider the green energy resources, an additional level of variability/uncertainty is added to the problem. I plan to extend the VNFs models and algorithms in a green perspective, considering multi-period problems and the energy consumption of the system as a principal objective.

6.2 Back to global optimization starting from process synthesis

Global optimization solution strategies can be divided in two main categories: exact and heuristic methods. Exact methods are essentially Branch-and-Bound (B&B) based procedures and, to be effectively implemented, ask for some information on the objective function and/or constraints (examples are Lipschitz-continuity, difference-of-convex, under-estimators, etc). Unfortunately, for many global optimization problems such kind of information is not enough to design an efficient exact strategy or, in most cases, it does not exist at all. In such cases, heuristic methods are used. Nevertheless, classical meta-heuristics (such as, random sampling, simulated annealing, etc) are not effective, if they are used in their standard form. It is essential to tailor meta-heuristics injecting some problem information. Indeed, some problem specific information can be available, even if it is not necessarily possible to express it in a mathematical form. The combination of "simple" global optimization strategies with the use of such informations proved to be very effective in many problems:

- For example, the protein-folding energetic landscape exhibits a funnel-like shape, that is a unimodal (or with few local minima) function perturbed by a large number of small oscillations (see Chapter 2). For this problem, and many others that exhibits a funnel-like shape, Monotonic Basin Hopping (MBH) is the core of most successful optimization methods. MBH combines a zero-temperature Simulated Annealing (or a Neighborhood Search heuristic) with the power of standard local optimization (e.g. Newton-like, sequential quadratic programming, trust-region). Essentially, it moves along the energetic landscape jumping from one local minimum to a closer (and lower) one, descending the funnel shape.

- To give another example, it is well-known that molecular clusters (as Morse or Lennard-Johns ones) have an almost spherical shape (and a funnel-like structure). An effective solution strategy is based on a modification of the standard MBH that uses a two-phase local optimization strategy (see Section 2.3.2): the first local optimization minimizes a penalty for clusters that are "too elongated", driving the search in the direction of more spherical solutions, the second one optimizes the original objective function.

The necessity to introduce information on solution strategies asks to global optimization practitioners to work with a multidisciplinary spirit. This mind-setting has two effects, from the one hand, allows to obtain specific information on the problem under study and allows to inject it in solution strategies, from the other hand, it gives the opportunity to tackle always new optimization problems arising from different application field.

New optimization problems offer the opportunity to develop new methods and, often, to transpose existing strategies on new applications. Adapting an existing method (for example the aforementioned two-phase local optimization) to a new problem is an interesting intellectual and scientific exercise, in fact it demands a process of abstraction/generalization, and it can be viewed as one of the process leading to new algorithms. For example, MBH, that was originally developed for solving molecular clusters optimization, was successfully used as a core-method for solving many other completely different problems, such as space-trajectory design problems. Thus, MBH can be now considered as a meta-heuristic. It is in this spirit that I want to pursuit my research on global optimization methods: starting from practical problems to derive more general ideas.

In the short period, I want to focus my research on process synthesis design problems, starting first to improve my work in membrane system design optimization, and then enlarge it to its general form, namely process synthesis: "Process synthesis is the assembly and interconnection of units into a process network - involving different physical and chemical phenomena to transform raw material and energy inputs into desired outputs - with the goal of optimizing economic, environmental, and/or social objectives" (from [45]). In general processes, the overall system can be composed of different interconnected sub-systems such as: distillation columns, chemical reactors, or a membrane process. Each of these sub-systems is, in turn, composed of sub-elements corresponding to physical devices (for example a membrane process can be composed of several membranes, compressors, etc.). I believe that this complex structure can be exploited to decompose also the resulting optimization problem allowing to solving it more effectively.

From an application point of view, process synthesis offers an interesting challenge: first of all, for their economical and environmental impact on industrial applications (CO_2 capture, air purification, etc), second, because even if there exist significant academic contributions on optimization-based process design, such contributions are not integrated in industrial practices, where simulation based methods and "by-hand" sensitivity analysis are still the main tool ([45]). This means that the existing methods proposed in the scientific literature do not respond to the needs of the (industrial) applications experts. This can be explained by the fact that such methods are too difficult to use outside academic and/or are not suitable to solve problems close enough to the industrial reality.

Membrane system design

In a first phase, I will focus on membrane system design problems, extending the work we already developed to obtain a more general and efficient solution method for a large spectrum of possible system configurations. The scientific challenge of proposing a so-

lution methodology that can solve the family of membrane system design problems is multiple:

- the underlying NLP is non-convex and performance constraints that are usually added to guarantee the quality levels necessary to industrial applications jeopardize the feasible region, making difficult even the convergence to local optimizers with the use of normally quite performing local solvers;
- defining the search space without loosing possibly optimal solutions (and without introducing not-physical designs) is still an open problem. For example, our work [J1] (see Chapter 3) put in evidence that not allowing self-loops (choice done by many researchers) removes some of the putative global optima of the process design;
- efficiently exploring the search space is quite challenging due to the presence of symmetries and the inherent combinatorial nature of the problem.

I will focus as core algorithm on Dissimilarity Population Based Methods (see [110] and Chapter 2), DPBM in what follows. Such methods are a natural extension of GO methods such a MBH or SA, where the problem presents a multi-funnel structure, or whenever the nature of the problem allows to "partition" the feasible solutions in categories using some global/structural information on the problem. Looking from far-away, these methods can be considered as the heuristic counter part of B&B methods. The different elements of the population represent different branches of the search tree. In B&B, a child node is a subproblem of the problem represented by the father node, a similar behavior is obtained enforcing similarity as a criterion for substituting elements of the children population to the parent population and therefore keeping each element of the population in a certain "search region". Eliminating elements from a population when they are not "good enough" with respect to the current population quality is a kind of heuristic bounding, allowing to keep the size of the population limited. A main difference between B&B and DPBM is represented by the way the space is partitioned, in classical B&B methods the search space is partitioned using explicitly the combinatorial (or geometrical) structure of the problem, and there is the guarantee that the sum of all subproblems contains all the feasible (or at least possibly optimal) solutions. In DPBM, the space is partitioned using problem features, that are usually a compact representation of the set of feasible (or good) solutions. As a consequence, differently from B&B, there is not necessarily a geometrical neighborhood between solutions of the same category, at least in the full variables space. These observations must be taken into account when the DPBM is designed and specialized for a given problem. Furthermore, they are also interesting to distinguish between DPBM and classical genetic algorithms (GA). I open this short parenthesis to highlight some differences, because in the community of process synthesis, and in particular in the membrane one, GA, usually coupled with simulation, are quite popular. Normally, classical GA substitute a child with one of its direct parent (if it improves with respect to the objective or some fitness measure) and do not search for a "closest" parent, like DPBM made. Therefore, they

can move from one funnel to another in one generation. Furthermore, GA, usually, use full representation of the variable space, and not condensed "features", that I believe is a key of the success of DPBM. Last, but not less important, the perturbation technique is different for the two methods, DPBM methods use perturbation strategies similar to MBH or SA (neighborhood search like), while GA uses two main techniques: mutation and cross-over, asking to determine a suitable encoding of the solutions to make the method effective. Mutation can be considered similar to neighborhood search, while cross-over is really specific to GA methods. Therefore, I believe, that even DPBM and GA shares some similarities, they can be considered two different meta-heuristics. The DPBM will be tailored to cope with the intrinsic combinatorial structure of the membrane design problem, using the different element of the population to explore different branching of the combinatorial problem. Starting from the problem considered in [J1], the number of degrees of freedom will be increased (completely variable pressures for each membrane, variable membrane permeability, addition of other devices, like heat-exchangers or expanders) to obtain a complete coverage of the industrial possible settings. To the best of our knowledge, it does not exist in the current state of the art any proposal of a solution method taking into account all these elements.

The contribution of this research it would not only to develop a solution algorithm for a large family of membrane systems, but also to reinforce some practices quite common in the global optimization community such as the use of benchmarking to compare different solution algorithms. In fact, in the process synthesis community is common practice to solve a different case-study in each research work (strategy completely justified by the different goals of the community, among which it is not central to validate optimization methods). I plan to devote a part of my research work to compare existing methods on a set of selected case-studies. I believe that this can be of general interest to start constructing a common ground to compare different solutions methods. I am aware that such a process has its own limits and risks, in fact benchmarking can became too pervasive and therefore all the community research can be drifted to adapt solution methods to the benchmark instead of solving "real" problems. Nevertheless, I believe that benchmarking is interesting to avoid falling in the opposite "trap", that is continouosly develop "new" methods to solve quite similar problems. Staying in this very line, I would like to spent part of my research to compare the two main optimization-based "philosophies" used in the process synthesis community : one is based on the use of full mathematical programming models to describe the overall system and applying some GO algorithm to them, the second is based on the coupling of simulation and optimization, the GO strategy works on a subset of the variables in a discretized way (configuration, membrane sizes, splits) and the resulting system is then simulated to obtain the overall system variables and the resulting objective function. Different GO methods are used with both strategies (and the full methods are validate on different case-studies), therefore, from the current state of the art, it is impossible to determine how these two approaches impact the overall optimization process. Thus, trying to assess in a systematic way the advantages and weaknesses of these two approaches is important to give insights on the future research directions on optimization-based membrane systems design.

Process synthesis

In a long term perspective, I aim at design solution methods for complex process synthesis. The experience gathered from membrane design will be helpful not only because membranes synthesis are often a module of a more complex process, and therefore the methods developed there can constitute elements of a more general algorithm, but also to invest more research time in the collaboration with the colleagues of the LRGP (Laboratory of Process Engineering), allowing to reinforce the common ground we started to construct[1]. In fact, as I already observed, the knowledge of the problem is of paramount importance to specialize GO algorithms and make them effective. In the current state of the art, different optimization approaches are applied to different scales of the system: from the abstract representation represented by the superstructure passing through rough representations to reach rigorous models, that are often discretization of the differential equation systems describing the underlying physical system ([45]). Each scale produces different models and asks for different optimization strategies. Some authors propose solution methods for some specific processes that integrate different scales models and solve the overall problem with a chain of optimization steps passing from a higher level model to a more refined one ([25]). This kind of approach pose its basis on the physical structure of the system and asks for two main steps: first, determine correct models for each layer [2] and solve them, second, device a overall strategy to connect results of one layer to another. As, general processes can be large scale and integrate different modular elements (membrane systems, distillation columns, etc), I think that multi-scale models are not enough to tackle large size and complex processes. A quite natural strategy is to approach the problem using decomposition methods. In particular, considering the process as composed of different elements or stages, a super-superstructure (or meta-model) can be imagined where different blocks represent functional units. Some research on this direction already exists that aims at determining superstructures using blocks for single functional units, as membranes, compressors, etc ([171], [134]).

The research lead I plan to follow to tackle process synthesis optimization shares some features with both multi-scale optimization and superstructure optimization strategies. In fact, I plan to propose a solution strategy that combines these two ideas to allow reuse all the already existing knowledge on solving some specific process synthesis problems and at the same time keep the size and complexity of the problem reasonable for the current state of the art solvers. Starting from specific case-studies, I want propose a solution method that considers both a decomposition of the system in sub-blocks and multi scale models for each sub-block. Differently from superstructure optimization methods, each sub-block will not be a single functional unit, but already a small process (for example a membranes systems inside a larger process). Furthermore, at least in the first stage, the "superstructure" will be user-defined and not the result of the optimization process as in the current state of the art works. Differently from the already proposed multi-scale strategies, that connect each scale model to the next one (linear connectivity), interconnection between sub-blocks, and possible within different scales, will be defined (grid

[1] The work presented in Chapter 3 is the result of this collaboration.

[2] the approximated models must not add stationary points that do not exist in the original problem

connectivity). This meta-model will be the base to build upon a global optimization strategy that can "have a view" of the overall optimization problem on different levels of details, both in a vertical direction (different problem scales) and in an horizontal direction (sub-blocks). Of course, the open challenge to design such an algorithm is huge, from determining effective multi-scale models to correctly solve and interconnect them. Furthermore, to be effective, the global optimization model must "autonomously" decide when to switch from a global to a local view of the system, and which scale use for each subproblem resolution. Using machine learning techniques to guide the search process seems a quite promising perspective, as already shown by some recent research both in global optimization ([42]) and in discrete optimization ([114]). I am aware, that this part of my research project is in an "embryonic state", nevertheless, I believe that starting from it as a mind-map will allow me to work on specific case-studies more effectively and, that this process will allow me to to bring these general ideas to maturity and to propose a new meta-algorithmic framework. The modular nature of such framework is suitable to build on it and propose a modular software that can be appealing from an industrial point of view. Indeed, as in many GO algorithm, standard local solvers are used to find local optima, in this general framework, other global optimization algorithms can be used to solve approximately each sub-system, allowing to exploit the power of existing tools.

Bibliography

[1] K.I. Aardal, M. Labbé, J. Leung, and M. Queyranne. *On the two-level uncapacitated facility location problem.* Universiteit Utrecht. UU-CS, Department of Computer Science. Utrecht University, Netherlands, 1995.

[2] O. Abdelkhalik and D. Mortari. N-impulse orbit transfer using genetic algorithms. *Journal of Spacecraft and Rockets*, 44:456–459, 2007.

[3] U.S. Environmental Protection Agency. Greenhouse gas equivalencies calculator. `http://www.epa.gov/cleanenergy/energy-resources/calculator.html`, July 2011.

[4] R. Agrawal and J. Xu. Gas-separation membrane cascades utilizing limited numbers of compressors. *AIChE Journal*, 42(8):2141–2154, 1996.

[5] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[6] Baris Aksanli, Jagannathan Venkatesh, Liuyi Zhang, and Tajana Rosing. Utilizing green energy prediction to schedule mixed batch and service jobs in data centers. In *Proceedings of the 4th Workshop on Power-Aware Computing and Systems*, HotPower '11, 2011.

[7] A. Aldraho and A. Kist. Enabling energy efficient and resilient networks using dynamic topologies. In *Sustainable Internet and ICT for Sustainability (SustainIT), 2012*, pages 1–8, 2012.

[8] A. Aliaga-Vicente, J. A. Caballero, and M. J. Fernandez-Torres. Synthesis and optimization of membrane cascade for gas separation via mixed-integer nonlinear programming. *AIChE Journal*, 63(6):1989–2006, 2017.

[9] Z. Allybokus, N. Perrot, J. Leguay, L. Maggi, and E. Gourdin. Virtual function placement for service chaining with partial orders and anti-affinity rules. *CoRR*, abs/1705.10554, 2017.

[10] E. Amaldi, A. Capone, S. Coniglio, and L.G. Gianoli. Energy-aware traffic engineering with elastic demands and MMF bandwidth allocation. In *2013 IEEE 18th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 169–174. IEEE, September 2013.

[11] Amazon Inc. AWS Elastic Beanstalk. http://aws.amazon.com/elasticbeanstalk/.

[12] A. Andrae and T. Edler. On global electricity usage of communication technology: Trends to 2030. *Challenges*, 6(1):117–157, apr 2015.

[13] D. Ardagna, S. Casolari, M. Colajanni, and B. Panicucci. Dual time-scale distributed capacity allocation and load redirect algorithms for cloud systems. *J. of Parallel and Distr. Computing*, 72(6):796 – 808, 2012.

[14] A. M. Arias, M. C. Mussati, P. L. Mores, N. J. Scenna, J. A. Caballero, and S. F. Mussati. Optimization of multi-stage membrane systems for CO2 capture from flue gas. *International Journal of Greenhouse Gas Control*, 53:371–390, 2016.

[15] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Commun. ACM*, 53(4):50–58, April 2010.

[16] A. Assad. Multicommodity network flows - a survey. *Networks*, 8(1):37–91, 1978.

[17] S. Avallone and G. Ventre. Energy efficient online routing of flows with additive constraints. *Computer Networks*, 56(10):2368–2382, July 2012.

[18] P. Avella, S. Mattia, and A. Sassano. Metric inequalities and the network loading problem. *Disc. Opt.*, 4:103–114, 2007.

[19] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba. On orchestrating virtual network functions. In *2015 11th International Conference on Network and Service Management (CNSM)*, pages 50–56, Nov 2015.

[20] Astrid Batterman, Joerg M. Gablonsky, Alton Patrick, Carl T. Kelley, Katheleen R. Kavanagh, Todd Coffey, and Cass T. Miller. Solution of a groundwater control

problem with implicit filtering. *Optimization and Engineering*, 3(2):189–199, June 2002.

[21] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[22] D. Bertsimas, D.B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011.

[23] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[24] R. Bhandari. *Survivable networks - Algorithms for diverse routing*. Kluwer, 1999.

[25] Lorenz T. Biegler, Yi dong Lang, and Weijie Lin. Multi-scale optimization for process systems engineering. *Computers & Chemical Engineering*, 60:17 – 30, 2014.

[26] D. Bienstock and S. Mattia. Using mixed-integer programming to solve power grid blackout problems. *Disc. Opt.*, 4:115–141, 2007.

[27] Robert Birke, Lydia Y. Chen, and Evgenia Smirni. Data centers in the cloud: A large scale performance study. In *IEEE CLOUD*, 2012.

[28] D. V. Boll, J. Donovan, R.L. Graham, and B.D. Lubachevsky. Improving dense packings of equal disks in a square. *The Electronic Journal of Combinatorics*, 7(R46):1–9, 2000.

[29] R. Bolla, R. Bruschi, A. Carrega, F. Davoli, D. Suino, C. Vassilakis, and A. Zafeiropoulos. Cutting the energy bills of internet service providers and telecoms through power management: An impact analysis. *Computer Networks*, 56(10):2320–2342, July 2012.

[30] Q. Botton, B. Fortz, L. Gouveia, and M. Poss. Benders decomposition for the hop-constrained survivable network design problem. *INFORMS J. Comput.*, 25:13–26, 2013.

[31] M. Bouet, J. Leguay, and V. Conan. Cost-based placement of vDPI functions in NFV infrastructures. In *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, pages 1–9, April 2015.

[32] R. Bounaceur, E. Berger, M. Pfister, A.A. Ramirez Santos, and E. Favre. Rigorous variable permeability modelling and process simulation for the design of polymeric membrane gas separation units: Memsic simulation tool. *Journal of Membrane Science*, 523:77–91, 2017.

[33] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[34] R. H. Byrd, J. Nocedal, and R. A. Waltz. Knitro: An integrated package for nonlinear optimization. In G. Di Pillo and M. Roma, editors, *Large-Scale Nonlinear Optimization*, pages 35–59. Springer US, Boston, MA, 2006.

[35] K. Lee C. Lee and S. Park. Benders decomposition approach for the robust network design problem with flow bifurcations. *Networks*, 62(1):1–16, 2013.

[36] Claudia Canali and Riccardo Lancellotti. Automated clustering of vms for scalable cloud monitoring and management. In *Software, Telecommunications and Computer Networks (SoftCOM), 2012 20th International Conference on*, 2012.

[37] Claudia Canali and Riccardo Lancellotti. Automated clustering of virtual machines based on correlation of resource usage. *Journal of Communications Software and Systems (JCOMSS)*, 8(4), jan. 2013.

[38] J.C. Cardona Restrepo, C.G. Gruber, and C. Mas Machuca. Energy profile aware routing. In *2009 IEEE International Conference on Communications Workshops*, pages 1–5. IEEE, June 2009.

[39] D. Carrera, M. Steinder, I. Whalley, J. Torres, and E. Ayguadé. Autonomic placement of mixed batch and transactional workloads. *IEEE Trans. Parallel Distrib. Syst.*, 23(2):219–231, 2012.

[40] L. G. Casado, I. Garcia, P. Szabö, and T. Csendes. Equal circles packing in square II: new results for up to 100 circles using the TAMSASS-PECS algorithm. In F. Giannessi, P.M. Pardalos, and T. Rapcsak, editors, *Optimization Theory: Recent developements from Mátraháza*, pages 207–224. Kluwer Academic Publishers, 1998.

[41] S. Casolari and M. Colajanni. Short-term prediction models for server management in internet-based contexts. *Decision Support Systems*, 48(1), 2009.

[42] A. Cassioli, D. Di Lorenzo, M. Locatelli, F. Schoen, and M. Sciandrone. Machine learning for global optimization. *Computational Optimization and Applications*, 51(1):279–303, Jan 2012.

[43] Ignacio Castillo, Frank J. Kampas, and Janos D. Pinter. Solving circle packing problems by global optimization: Numerical results and industrial applications. *European Journal of Operational Research*, to appear, 2006.

[44] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power awareness in network design and routing. In *2008 IEEE INFOCOM - The 27th Conference on Computer Communications*, pages 457–465. IEEE, April 2008.

[45] Qi Chen and I.E. Grossmann. Recent developments and challenges in optimization-based process synthesis. *Annual Review of Chemical and Biomolecular Engineering*, 8(1):249–283, 2017. PMID: 28375774.

[46] L. Chiaraviglio, A. Cianfrani, A. Coiro, M. Listanti, and M. Polverini. Green routing switching and transport. In K. Samdanis, P. Rost, A. Maeder, M. Meo, and C. Verikoukis, editors, *Green Communications: Principles, Concepts and Practice*, chapter 13, pages 257–276. John Wiley and Sons, Ltd, 2015.

[47] L. Chiaraviglio, D. Ciullo, M. Mellia, and M. Meo. Modeling sleep modes gains with random graphs. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 355–360. IEEE, April 2011.

[48] L. Chiaraviglio, D. Ciullo, M. Mellia, and M. Meo. Modeling sleep mode gains in energy-aware networks. *Computer Networks*, 57(15):3051–3066, October 2013.

[49] L. Chiaraviglio, M. Mellia, and F. Neri. Energy-aware networks: Reducing power consumption by switching off network elements. In *FEDERICA-Phosphorus tutorial and workshop (TNC2008)*. Citeseer, 2008.

[50] L. Chiaraviglio, M. Mellia, and F. Neri. Energy-aware backbone networks: A case study. In *2009 IEEE International Conference on Communications Workshops*, pages 1–5. IEEE, June 2009.

[51] L. Chiaraviglio, M. Mellia, and F. Neri. Reducing power consumption in backbone networks. In *2009 IEEE International Conference on Communications*, pages 1–6. IEEE, June 2009.

[52] L. Chiaraviglio, M. Mellia, and F. Neri. Minimizing ISP Network Energy Cost: Formulation and Solutions. *IEEE/ACM Transactions on Networking*, 20(2):463–476, April 2012.

[53] T. D. Choi and Carl T. Kelley. Superlinear convergence and implicit filtering. *SIAM Journal on Optimization*, 10(4):1149–1162, 2000.

[54] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. Maestro. Ieee 802.3az: the road to energy efficient ethernet. *IEEE Communications Magazine*, 48(11):50–56, November 2010.

[55] I. Contreras and E. Fernández. General network design: A unified view of combined location and network design problems. *European Journal of Operational Research*, 219(3):680 – 697, 2012. Feature Clusters.

[56] D. Coudert, A.M. Koster, T.K. Phan, and M. Tieves. Robust redundancy elimination for energy-aware routing. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pages 179–186. IEEE, August 2013.

[57] H.G. Croft, K.J. Falconer, and R.K. Guy. *Unsolved problems in Geometry*. Springer-VErlang, New York, 1991.

[58] B. Dachwald. Optimization of interplanetary solar sailcraft trajectories using evolutionary neurocontrol. *AIAA Journal of Guidance, Control, and Dynamics*, 27:66–72, 2004.

[59] C. de Groot, R. Peikert, D. Würtz, and M. Monagan. Packing circles in a square: a review and new results. In *System Modelling and Optimization*, pages 45–54, Zürich, 1991. Proc. 15th IFIP Conf.

[60] P. Di Lizia and G. Radice. Advanced global optimization tools for mission analysis and design. Technical Report 03-4101b, ESA, 2004.

[61] Artan Dimnaku, Rex Kincaid, and Michael W. Trosset. Approximate solutions of continuous dispersion problems. *Annals of Operations Research*, 136(1):65–80, Apr 2005.

[62] L. C. W. Dixon. Global optima without convexity. In H. Greenberg, editor, *Design and Implementation Optimization Software*, pages 449–479. Sijthoff and Noordhoff, Alphen aan den Rijn, 1978.

[63] Elizabeth D. Dolan and Jorge J. Morè. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91:201–213, 2002.

[64] Aleksandar Donev, Salvatore Torquato, Frank H. Stillinger, and Robert Connelly. A linear programming algorithm to test for jamming in hard-sphere packings. *Journal of Computational Physics*, 197(1):139 – 166, 2004.

[65] Jonathan P. K. Doye, Robert H. Leary, Marco Locatelli, and Fabio Schoen. Global optimization of Morse clusters by potential energy transformations. *INFORMS Journal On Computing*, 16:371–379, 2004.

[66] G. Scott Graham E. D. Lazowska, J. Zahorjan and K. C. Sevcik. *Quantitative System Performance, Computer System Analysis Using Queueing Network Models.* Prentice-Hall, 1984.

[67] M. M. Faruque Hasan, R. C. Baliban, J. A. Elia, and C. A. Floudas. Modeling, simulation, and optimization of postcombustion co_2 capture for variable feed concentration and flow rate. 1. chemical absorption and membrane processes. *Industrial and Engineering Chemistry Research*, 51(48):15642–15664, 2012.

[68] E. Favre. Comprehensive membrane science and engineering. In E. Drioli and L. Giorno, editors, *Polymeric membranes for gas separations*, volume 2, chapter 10, pages 155–211. Elsevier Science, Amsterdam, 2010.

[69] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach. Virtual network embedding: A survey. *IEEE Communications Surveys Tutorials*, 15(4):1888–1906, Fourth 2013.

[70] M. Florian. Nonlinear cost network models in transportation analysis. In *Netflow at Pisa*, volume 26 of *Mathematical Programming Studies*, pages 167–196. Springer Berlin Heidelberg, 1986.

[71] Christodoulos A. Floudas. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. Oxford University Press, New York, 1995.

[72] Ulrich Focken, Matthias Lange, and Hans-Peter Waldl. Previento - A Wind Power Prediction System with an Innovative Upscaling Algorithm. In *Proc. of the 2002 Global Windpower Conference*, SIGMETRICS '12, 2012.

[73] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 519–528. IEEE, 2002.

[74] GreenTouch Foundation. Reducing the Net Energy Consumption in Communications Networks by up to 98% by 2020 , aug 2015.

[75] F. Francois, K. Moessner, and S. Georgoulas. Optimizing link sleeping reconfigurations in isp networks with off-peak time failure protection. *IEEE Transactions on Network and Service Management*, 10(2):176–188, June 2013.

[76] F. Francois, N. Wang, K. Moessner, and S. Georgoulas. Optimization for time-driven link sleeping reconfigurations in ISP backbone networks. In *Proceedings of the 2012 IEEE Network Operations and Management Symposium, NOMS 2012*, pages 221–228. IEEE, April 2012.

[77] P. Gabrielli, M. Gazzani, and M. Mazzotti. On the optimal design of membrane-based gas separation processes. *Journal of Membrane Science*, 526:118–130, 2017.

[78] P.J. Gage, R.D. Braun, and I.M. Kroo. Interplanetary trajectory optimization using a genetic algorithm. *Journal of the Astronautical Sciences*, 43:59–75, 1995.

[79] M. Galizia, W. S. Chi, Z. P. Smith, T. C. Merkel, R. W. Baker, and B. D. Freeman. Polymers and mixed-matrix membranes for gas and vapor separation: A review and prospective opportunities. *Macromolecules*, 50(20):7809–7843, 2017.

[80] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[81] R.G. Garroppo, S. Giordano, G. Nencioni, and M.G. Scutellà. Mixed integer non-linear programming models for green network design. *Computers & Operations Research*, 40(1):273–281, January 2013.

[82] P. E. Gill, W. Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12:979–1006, 2002.

[83] P. C. Gilmore and Carl T. Kelley. An implicit filtering algorithm for optimization of functions with many local minima. *SIAM Journal on Optimization*, 5(2):269–285, 1995.

[84] R.L. Graham and B.D. Lubachevsky. Repeated patterns of dense packings of equal disks in a square. *The Electronic Journal of Combinatorics*, 3:1–16, 1996.

[85] GreenTouch consortium. www.greentouch.org.

[86] A. Grosso, A. R. Jamali, M. Locatelli, and F. Schoen. Solving the problem of packing equal and unequal circles in a circular container. *J. of Global Optimization*, 47(1):63–81, May 2010.

[87] Andrea Grosso, Marco Locatelli, and Fabio Schoen. A population-based approach for hard global optimization problems based on dissimilarity measures. *Mathematical Programming*, 110(2):373–404, Jul 2007.

[88] GT-ITM. Georgia Tech Internetwork Topology Models. `http://www.cc.gatech.edu/fac/Ellen.Zegura/gt-itm/`.

[89] R. Guerzoni, R. Trivisonno, I. Vaishnavi, Z. Despotovic, A. Hecker, S. Beker, and D. Soldani. A novel approach to virtual networks embedding for SDN management and orchestration. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–7, May 2014.

[90] J.T. Haahr, T.R. Stidsen, and M. Zachariasen. Heuristic methods for shared backup path protection planning. In *Proc. of RNDM 2012*, pages 712–718, 2012.

[91] R. Hays. Active/idle toggling with 0base-x for energy efficient ethernet. presentation to the IEEE 802.3az Task Force, Nov. 2007, 2007.

[92] H. Huang, W. Huang, Q. Zhang, and D. Xu. An improved algorithm for the packing of unequal circles within a larger containing circle. *European Journal of Operational Research*, 141:440–453, 2002.

[93] W. Huang, Y. Li, B. Jurkowiak, C.M. Li, and R.C. Xu. A two-level search strategy for packing unequal circles into a circular container. In *Proceedings of the International Conference on Principles and Practice of Constraint Programming*, pages 868–872. Springer-Verlag, 2003.

[94] J. Hwang, K. K. Ramakrishnan, and T. Wood. Netvm: High performance and flexible networking using virtualization on commodity platforms. *IEEE Transactions on Network and Service Management*, 12(1):34–47, March 2015.

[95] F. Idzikowski, L. Chiaraviglio, R. Duque, F. Jimenez, and E. Le Rouzic. Green horizon: Looking at backbone networks in 2020 from the perspective of network operators. In *2013 IEEE International Conference on Communications (ICC)*, pages 4455–4460. IEEE, June 2013.

[96] Dario Izzo. Advances in global optimisation for space trajectory design. In *25th International Symposium on Space Technology and Science*. Japan Society for Aeronautical and Space and ISTS, 2006.

[97] Dario Izzo. 1st ACT global trajectory optimisation competition: Problem description and summary of the results. *Acta Astronautica*, 61(9):731–734, November 2007.

[98] Dario Izzo, V. M. Becerra, D. R. Myatt, S. J. Nasuto, and J. M. Bishop. Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories. *Journal of Global Optimization*, 38:283–296, 2007.

[99] J. Kennington, E. Olinick, and G. Spiride. Basic mathematical programming models for capacity allocation in mesh-based survivable networks. *Omega*, 35:629–644, 2007.

[100] H. Kerivin and A. Mahjoub. Design of survivable networks: a survey. *Networks*, 46(1):1–21, 2005.

[101] Y.H. Kim and D.B. Spencer. Optimal spacecraft rendezvous using genetic algorithms. *Journal of Spacecraft and Rockets*, 39:859–865, 2002.

[102] J. Koomey. Growth in data center electricity use 2005 to 2010. Technical report, Oakland, CA: Analytics Press, August 2011.

[103] C. Lange, D. Kosiankowski, A. Betker, H. Simon, N. Bayer, D. von Hugo, H. Lehmann, and A. Gladisch. Energy Efficiency of Load-Adaptively Operated Telecommunication Networks. *Journal of Lightwave Technology*, 32(4):571–590, feb 2014.

[104] Robert H. Leary. Global optimization on funneling landscapes. *Journal of Global Optimization*, 18:367–383, 2000.

[105] Julian Lee, In-Ho Lee, and Jooyoung Lee. Unbiased global optimization of lennard-jones clusters for $n \leq 201$ using the conformational space annealing method. *Phys. Rev. Lett.*, 91:080201, Aug 2003.

[106] G. Lin, S. Soh, M. Lazarescu, and K.W. Chin. Power-aware routing in networks with delay and link utilization constraints. In *Proceedings - Conference on Local Computer Networks, LCN*, pages 272–275. Ieee, October 2012.

[107] G. Lin, S. Soh, M. Lazarescu, and K.W. Chin. Power-aware routing in networks with quality of services constraints. *Transactions on Emerging Telecommunications Technologies*, pages n/a–n/a, March 2014.

[108] Zhenhua Liu, Yuan Chen, Cullen Bash, Adam Wierman, Daniel Gmach, Zhikui Wang, Manish Marwah, and Chris Hyser. Renewable and cooling aware workload management for sustainable data centers. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '12, 2012.

[109] M. Locatelli. On the multilevel structure of global optimization problems. *Computational Optimization and Applications*, 30(1):5–22, Jan 2005.

[110] M. Locatelli and F. Schoen. *Global Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2013.

[111] Marco Locatelli. On the multilevel structure of global optimization problems. *Computational Optimization and Applications*, 30(1):5–22, January 2005.

[112] Marco Locatelli and Ulrich Raber. Packing equal circles in a square: a deterministic global optimization approach. *Discrete Applied Mathematics*, 122:139–166, 2002.

[113] Marco Locatelli and Fabio Schoen. Efficient algorithms for large scale global optimization: Lennard-jones clusters. *Computational Optimization and Applications*, 26(2):173–190, Nov 2003.

[114] Andrea Lodi and Giulia Zarpellon. On learning and branching: a survey. *TOP*, 25(2):207–236, Jul 2017.

[115] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary. Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 98–106, May 2015.

[116] Marcelo Caggiani Luizelli, Weverton Luis da Costa Cordeiro, Luciana S. Buriol, and Luciano Paschoal Gaspary. A fix-and-optimize approach for efficient and large scale virtual network function placement and chaining. *Computer Communications*, 102:67 – 77, 2017.

[117] T. Lukovszki and S. Schmid. Online admission control and embedding of service chains. In *Post-Proceedings of the 22Nd International Colloquium on Structural Information and Communication Complexity - Volume 9439*, SIROCCO 2015, pages 104–118, New York, NY, USA, 2015. Springer-Verlag New York, Inc.

[118] B. Luo, W. Liu, and A. Al-Anbuky. Energy aware survivable routing approaches for next generation networks. In *2013 Australasian Telecommunication Networks and Applications Conference (ATNAC)*, pages 160–165. IEEE, November 2013.

[119] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan. A power benchmarking framework for network devices. *NETWORKING 2009*, pages 795–808, 2009.

[120] Mihály Csaba Markót and Tibor Csendes. A new verified optimization technique for the "packing circles in a unit square" problem. *SIAM J. on Optimization*, 16:193–219, 2005.

[121] S. Mattia. The robust network loading problem with dynamic routing. *Comput. Optim. Appl.*, 54(3):619–643, 2012.

[122] S. Mattia. Solving survivable two-layer network design problems by metric inequalities. *Comput. Optim. Appl.*, 51(2):809–834, 2012.

[123] S. Mehraghdam, M. Keller, and H. Karl. Specifying and placing chains of virtual network functions. In *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, pages 7–13, Oct 2014.

[124] H. Mellah and B. Sansò. Review of facts, data and proposals for a greener internet. In *Proceedings of the 6th International ICST Conference on Broadband Communications, Networks and Systems*. IEEE, 2009.

[125] T.C. Merkel, H. Lin, X. Wei, and R. Baker. Power plant post-combustion carbon dioxide capture: an opportunity for membranes. *Journal of Membrane Science*, 359:126–139, 2010.

[126] T.C. Merkel, M. Zhou, and R. Baker. Carbon dioxide capture with membranes at an igcc power plant. *Journal of Membrane Science*, 389:441–450, 2012.

[127] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and S. Davy. Design and evaluation of algorithms for mapping and scheduling of virtual network functions. In *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, pages 1–9, April 2015.

[128] R. Miller. Google Data Center Frequently Asked Questions. `http://www.datacenterknowledge.com/archives/2008/03/27/google-data-center-faq/`, March 2008.

[129] R. Miller. Google's chiller-less dc. `http://www.datacenterknowledge.com/archives/2009/07/15/googles-chiller-less-data-center`, July 2009.

[130] M. Minami and H. Morikawa. Some open challenges for improving the energy efficiency of the internet. In *Proc. 3rd International Conference on Future Internet (CFI 2008)*, 2008.

[131] B. Mumey, J. Tang, and S. Hashimoto. Enabling green networking with a power down approach. In *2012 IEEE International Conference on Communications (ICC)*, pages 2867–2871. IEEE, June 2012.

[132] D. R. Myatt, V. M. Becerra, S. J. Nasuto, and J. M. Bishop. Advanced global optimisation for mission analysis and design. Technical Report 18138/04/NL/MV, ESA, 2004.

[133] Tran Manh Nam, Nguyen Huu Thanh, Hoang Trung Hieu, Nguyen Tien Manh, Nguyen Van Huynh, and Hoang Duong Tuan. Joint network embedding and server consolidation for energy-efficient dynamic data center virtualization. *Computer Networks*, 125:76 – 89, 2017. Softwarization and Caching in NGN.

[134] Thibaut Neveux. Ab-initio process synthesis using evolutionary programming. *Chemical Engineering Science*, 185:209 – 221, 2018.

[135] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, USA, second edition, 2006.

[136] K.J. Nurmela and P.R.J. Oestergard. Packing up to 50 equal circles in a square. *Discrete Computational Geometry*, 18:111–120, 1997.

[137] K.J. Nurmela and P.R.J. Oestergard. More Optimal Packings of Equal Circles in a Square. *Discrete Computational Geometry*, 22:439–457, 1999.

[138] B. Ohs, J. Lohaus, and M. Wessling. Optimization of membrane based nitrogen removal from natural gas. *Journal of Membrane Science*, 498:291–301, 2016.

[139] A. D. Olds, C. A. Kluever, and M.L. Cupples. Interplanetary mission design using differential evolution. *Journal of Spacecraft and Rockets*, 44:1060–1070, 2007.

[140] S. Orlowski and M. Pióro. Complexity of column generation in network design with path-based survivability mechanisms. *Networks*, 59(1):132–147, 2012.

[141] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski. SNDlib 1.0 – Survivable Network Design library. *Networks*, 55(3):276–286, May 2010.

[142] J.D. Pinter and F.J. Kampas. Nonlinear Optimization in Mathematica using Math-Optimizer Professional. *Mathematica in Education and Research*, 10:1–18, 2005.

[143] M. Pióro and D. Medhi. *Routing, Flow and Capacity Design in Communication and Computer Networks*. Morgan Kaufman, 2004.

[144] R. Qi and M. A. Henson. Optimization-based design of spiral-wound membrane systems for CO2/CH4 separations. *Separation and Purification Technology*, 13:209–225, 1998a.

[145] R. Qi and M. A. Henson. Optimal design of spiral-wound membrane networks for gas separations. *Journal of Membrane Science*, 148:71–89, 1998b.

[146] R. Qi and M. A. Henson. Membrane system design for multicomponent gas mixtures via mixed-integer nonlinear programming. *Computers and Chemical Engineering*, 24:2719–2737, 2000.

[147] A. A. Ramirez-Santos, C. Castel, and E. Favre. Utilization of blast furnace flue gas: Opportunities and challenges for polymeric membrane gas separation processes. *Journal of Membrane Science*, 526:191–204, 2017.

[148] G. Rauwolf and V. Coverstone-Carroll. Near-optimal low-thrust orbit transfers generated by a genetic algorithm. *Journal of Spacecraft and Rockets*, 33:859–862, 1996.

[149] J. Rolia, L. Cherkasova, and C. McCarthy. Configuring Workload Manager Control Parameters for Resource Pools. In *10th IEEE NOMS*, Vancouver, Canada, April 2006.

[150] M. Scholz, M. Alders, T. Lohaus, and M. Wessling. Structural optimization of membrane-based biogas. *Journal of Membrane Science*, 474:1–10, 2015.

[151] A. Shafiee, M. Nomvar, Z. Liu, and A. Abbas. Automated process synthesis for optimal flowsheet design of a hybrid membrane cryogenic carbon capture process. *Journal of Cleaner Production*, 150:309–323, 2017.

[152] J. Soares, M. Dias, J. Carapinha, B. Parreira, and S. Sargento. Cloud4NFV: A platform for virtual network functions. In *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, pages 288–293, Oct 2014.

[153] E. Specht. Packomania. `http://www.packomania.com`. Accessed: 2018-09-06.

[154] R. Spillman. Economics of gas separation membrane processes. In R. D. Noble and S. A. Stern, editors, *Membrane Separations Technology. Principles and Applications*, pages 589–667. Elsevier Science, Amsterdam, 1995.

[155] M. Stoer and G. Dahl. A polyhedral approach to multicommodity survivable network design. *Numer. Math.*, 68(1):149–167, 1994.

[156] Y.G. Stoyan and G.N. Yaskow. Mathematical model and solution method of optimization problem of placement of rectangles and circles taking into account special constraints. *International Transactions in Operational Research*, 5:45–57, 1998.

[157] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici. A scalable application placement controller for enterprise data centers. In *WWW2007*, 2007.

[158] M. Ungerank, G. Baumgarten, M. Priske, and H. Roegl. Process for separation of gases. WO Patent WO 2012/000727, 2012.

[159] R. V. S. Uppaluri, P. Linke, and A. C. Kokossis. Synthesis and optimization of gas permeation membrane networks. *Industrial and Engineering Chemistry Research*, 43:4305–4322, 2004.

[160] R. V. S. Uppaluri, R. Smith, P. Linke, and A. C. Kokossis. On the simultaneous optmization of pressure and layout for gas permeation membrane systems. *Journal of Membrane Science*, 280:832–848, 2006.

[161] W. Van Heddeghem, F. Idzikowski, E. Le Rouzic, J.Y. Mazeas, H. Poignant, S. Salaun, B. Lannoo, and D. Colle. Evaluation of power rating of core network equipment in practical deployments. In *2012 IEEE Online Conference on Green Communications (GreenCom)*, pages 126–132. IEEE, September 2012.

[162] N. Vasić and D. Kostić. Energy-aware traffic engineering. In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking - e-Energy '10*, page 169, New York, New York, USA, 2010. ACM Press.

[163] M. Vasile and M. Locatelli. A hybrid multiagent approach for global trajectory optimization. *Journal of Global Optimization*, 2008. to appear.

[164] M. Vasile, E. Minisci, and M. Locatelli. On testing global optimization algorithms for space trajectory design. In *Proceedings of AIAA 2008*, 2008.

[165] Massimiliano Vasile. A global approach to optimal space trajectory design. *Advances in the Astronautical Sciences*, 114:629–647, February 2003.

[166] Massimiliano Vasile. Design of Earth-Mars transfer trajectories using evolutionary-branching technique. *Acta Astronautica*, 56:705–720, 2005.

[167] Massimiliano Vasile and Paolo De Pascale. Design of Earth-Mars transfer trajectories using evolution-branching technique. In *54th International Astronautical Congress*, October 2003.

[168] Tamás Vinkó, Dario Izzo, and Claudio Bombardelli. Benchmarking different global optimisation techniques for preliminary space trajectory design. In *58th International Astronautical Congress*. International Astronautical Federation (IAF), September 2007.

[169] D. J. Wales and J. P. K. Doye. Global optimization by Basin-Hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *Journal of Physical Chemistry A*, 101(28):5111–5116, 1997.

[170] David J. Wales. Energy landscapes and structure prediction using basin-hopping. In *Modern Methods of Crystal Structure Prediction*, chapter 2, pages 29–54. Wiley-Blackwell, 2010.

[171] Ligang Wang, Philip Voll, Matthias Lampe, Yongping Yang, and André Bardow. Superstructure-free synthesis and optimization of thermal power plants. *Energy*, 91:700 – 711, 2015.

[172] N. Wang, K. Ho, G. Pavlou, and M. Howarth. An overview of routing optimization for internet traffic engineering. *IEEE Communications Surveys & Tutorials*, 10(1):36–56, 2008.

[173] A. Wolke and G. Meixner. Twospot: A cloud platform for scaling out web applications dynamically. In *ServiceWave*, 2010.

[174] Y. Wu, B. Guo, Y. Shen, J. Wang, and X. Liu. Toward energy-proportional internet core networks: an energy-minimized routing and virtual topology design for internet protocol layer. *International Journal of Communication Systems*, pages n/a–n/a, October 2013.

[175] D. Yuan. *Optimization models and methods for communication network design and routing*. PhD thesis, Linköping Uni- versity, 2001.

[176] M. Yuan, K. Narakornpijit, R. Haghpanah, and J. Wilcox. Consideration of a nitrogen-selective membrane for post combustion. *Journal of Membrane Science*, 465:177–184, 2014.

[177] D. Zhang and A. Deng. An effective hybrid algorithm for the problem of packing circles into a larger containing circle. *Computers & Operations Research*, 32:1941–1951, 2005.

[178] M. Zhang, C. Yi, B. Liu, and B. Zhang. Greente: Power-aware traffic engineering. In *The 18th IEEE International Conference on Network Protocols*, pages 21–30. IEEE, October 2010.

[179] X. Zhu, D. Young, B. Watson, Z. Wang, J. Rolia, S. Singhal, B. McKee, C. Hyser, D.Gmach, R. Gardner, T. Christian, and L. Cherkasova:. 1000 islands: An integrated approach to resource management for virtualized data centers. *Journal of Cluster Computing*, 12(1):45–57, 2009.

[180] A. Zimmermann. Circle packing contest. `http://www.recmath.org/contest/CirclePacking/index.php`. Accessed: 2018-09-06.

7 Curriculum Vitae

7.1 Contact, Current Position and Personal Informations

Current Position: Associate Professor (Maitre de conferences, tenured)

Email:bernardetta.addis@loria.fr
Tel: +33 (0)3.83.54.95.20.98
Office: C031
LORIA
615 rue du Jardin Botanique
54600 Villers les Nancy
France

Personal Information:
Date of Birth: 28th February 1976
Place of Birth: Firenze, Italia
Citizenship: Italian
Children: 1 (28/12/2016)

7.2 Education

October 2005 Advanced School on Parallel Computing at CINECA (Italian center for High Performance Computing)

July 2005 Summer School on Parallel Computing at CINECA

April 2005 Ph.D. in Computer Science and Automation Engineering ("Ingegneria Informatica e dell'Automazione"),
Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze (Italy)
Title: "Global optimization using local searches"
Supervisor: Prof. Fabio SCHOEN
Members of the Jury: Alessandro AGNETIS, M. LA CAVA, E. GRAZZINI
Date and place of dissertation: 27/04/2005, University of Florence

April-August 2004 Thesis Part Appointment at Mathematics and Computer Science Division - Argonne National Laboratory (Chicago, IL,USA)
Supervisor: Sven LEYFFER (Computational Mathematician).

January 2002-December 2004 Ph.D. program in (Computer Science and Automation Engineering) at Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze.

April 2001 Master degree (cum laude) in Computer Science Engineering
Thesis title: "Determination of Biomolecular docking by means of numerical optimization algorithms"
Supervisor: prof. Fabio SCHOEN

7.3 Work experience

Start	End	Institutions	Positions and status
09/2013	now	Université de Lorraine (ENSMN) - LORIA	Assistant professor
12/2010	08/2013	Dipartimento di Informatica Università degli Studi di Torino, Italy	Research Associate
05/2007	11/2010	Dipartimento di Elettronica e Informazione (D.E.I. e Bioingegneria) Politecnico di Milano, Italy	Research Associate
02/2006	04/2007	Dipartimento di Ingegneria dell'Informazione Università degli Studi di Siena, Italy	Research Fellow
01/2005	01/2006	Dipartimento di Sistemi e Informatica Università degli Studi di Firenze, Italy	Research Fellow
05/2001	12/2001	Dipartimento di Sistemi e Informatica Università degli Studi di Firenze, Italy	Research Engineer

For accademic year 2017/2018 I had a "delegation CNRS" and for the first semester of accademic year 2018/2017 a "CRCT" (both are special "grants" allowing to have full time research periods without teaching).

7.4 Research periods spent abroad

30 January-16 February 2018 Research activity at Politecnico di Milano (working with Giuliana Carello) and University of Rome "Tor Vergata" (working with Veronica Piccialli)
1-11 July 2014 Research activity at Gerad, (Groupe d'études et de recherche en analyse des décisions), Montreal (working with Prof. Brunilde Sansò)
25 October-15 November 2012 Research activity at Gerad, (Groupe d'études et de recherche en analyse des décisions), Montreal (working with Proff. Brunilde Sansò and Patrick Soriano)
26 April-6 May 2011 Research activity at Gerad (Groupe d'études et de recherche en analyse des décisions), Montreal (working with Prof. Brunilde Sansò)
15-19 February 2010 Research activity at ACT-ESA (Advanced Concept Team - European Space Agency)
November 2004 Research activity at Department of Statistics and Decision Support

Systems, University of Wien (working with Proff. Werner Schachinger and Immanuel M. Bomze)

7.5 Prizes and awards

Personal Grants

- 2016: PEDR (Prime Encadrement Doctorale et Recherche).

Optimization Competitions and putative optima databases
For many global optimization problems, certifying optimality is not possible. It is common practice, to maintain databases where current putative optima (best solutions found up to now) for some optimization problems are kept to allow researchers to compare on a common ground. In the following, I indicate links to website containing putative optima (the name of the specific problem is reported in parenthesis):

- `http://www.esa.int/gsp/ACT/inf/projects/gtop/gtop.html`
  MGA-1DSM: Tandem, Messenger reduced, Cassini 2, Rosetta
- `http://www.packomania.com/`
  circles in the unit square, unequal circles in a circle (see Circle packing contest below)

Furthermore, sometimes international competitions (this practice is common also in other fields of optimization) are dedicated to open optimization problems. Usually the problem is disclosed at some date and closed after one or more months, and the ranking is based on the quality of solutions sent by participants:

- October 2010. GTOC5 (Global Trajectory Optimization Competition) partecipation ESA-ACT team (European Space Agengy - Advanced Concept Team) - fourth place (see [J10] for details on the solution method).
- January 2006. Al Zimmermann's Programming Contests: Circle Packing (pack n non-overlapping discs with radii from 1 to n into as small a circle as possible) - Winner with F. Schoen and M. Locatelli (see [J20] for details on the solution method).

Funded projects as Principal Investigator

PGMO 2018 - IMPRESS the Impact of optiMization in PRocESs Synthesis. Interdisciplinary project in collaboration with LRGP laboratory and EDF (10000 eur for 1 year).

PEPS Mirabelle 2016 - MIND Mixed Integer Nonlinear programming for membrane system Design. Interdisciplinary project in collaboration with LRGP laboratory (7500 eur for 1 year).

June-September 2015 Industrial research project with Pharmagest. Restructuring of the optimization engine for an order planning software for pharmacies (6000 eur - co-responsible with W. Ramdane Cherif).

Paper awards

- [J14] was one of 5 most highly cited papers in DAMS from 2014 to 2016.
- [C1] received the "Best paper award" in the NTMS 2016 conference.

7.6 Supervision of research activities

PhD supervision

Active participation to scientic supervision before permanent position:

- 2011-2014: Luca Giovanni Gianoli – PhD student at Polimi-Polytech Montreal.
  Supervisors: Brunilde SANSO, Antonio CAPONE.
  Title: Energy-aware traffic engineering for wired IP networks.

  My contribution focused (in collaboration with G. Carello) on the mathematical modelling and optimization algorithms (and the computational results analysis) for all the papers I coautored with Luca. Furthermore, we supervised and gave a large contribution to all the papers reduction, and in particular to the organization of the material for the survey we cohautored.

Thesis at UL with official autorization to co-direction (Autorization á Codiriger Thése):

- 2013-2017: Evangelia TSIONTSIOU – PhD student at LORIA, UL - regional project SATELOR.
  Supervisor: Ye-Qiong SONG.
  Title: Multiconstrained QoS routing for wireless sensor networks with application to smart space for ambient assisted living.

  The thesis of Evangelia is the first one I officially supervised (the same year I arrived at Loria). The general subject of the thesis was proposed by prof. SONG. My main supervision contribution was on the work developed in the second and third year of thesis, where a optimization-based routing protocol to extend the network lifetime was proposed. My contribution extends to supporting the final manuscript organization and editing.

- 2015-2018: Meihui GAO – PhD Student LORIA, UL - supported by a ministerial Grant.
  Supervisor: Ye-Qiong SONG.
  Title: Optimization models and methods for Network Functions Virtualization architectures.

I proposed the subject of the PhD thesis, and supervised all the work of Meihui GAO. Meihui spent some months in Italy under the supervision of G. Carello. The thesis will be discussed at the end of the year.

- 2016-2019: Marjan BOZORG – PhD Student LRGP-LORIA, UL and Univ. of Rome "Tor Vergata" - supported by a ministerial Grant and a Vinci grant for Italian-French mobility).
  Supervisors: Christophe CASTEL (LRGP), Veronica PICCIALLI (Rome).
  Title: Optimization of the design of membrane processes.

  The subject of this PhD thesis is the result of a multisciplinary collaboration with the LRGP laboratory. V. Piccialli and me supervise the optimization methods contribution, while C. Castel covers the application expertize. I am also in charge of the coordination of the work between the different participant of the research (see Mirabelle Project MIND).

Recent Master Students supervision

- 2017 Francesca DE BETTIN (M2, Politecnico Milano, Italy). Supervisor: Giuliana CARELLO. Title: On a Network Function Virtualization Orchestration problem with uniform chain allocation and simple path routing: properties and formulation.
- 2016 Chiara Antonella RAFFAELLI (M2, Politecnico Milano, Italy). Supervisor: Giuliana CARELLO. Title: Joint Energy Management of cloud infrastructures and networks.
- 2015 Magdalena KRZACZKOWSKA (M2, Univ. of Rome, "Tor Vergata" - stage at LORIA). Supervisor: Veronica PICCIALLI. Title: An optimization framework for membrane system automatic parameter tuning.
- 2015 Giulia LA ROSA (M2, Politecnico di Milano). Supervisors: Giuliana CARELLO, Antonio CAPONE. Title: Optimizing cellular networks planning and management from an energy-aware perspective: MILP formulations and MILP-based techniques.
- 2015 Mattia RACO (M2, Politecnico di Milano). Supervisor: Giuliana CARELLO. Title: Heuristics for an energy-aware management problem in cellular networks.
- 2008 Fosco Angelo Bombardieri (M2, Politecnico di Milano). Supervisor: Danilo ARDAGNA. Title: Tecniche di Resource Allocation per la gestione dei consumi energetici di Service Center con vincoli di availability (Resource Allocation methods for service center energy consumption with availability constraints).

7.7 Responsibilities

7.7.1 Editorial and reviewing activity

Member of the conference/workshop program committees

- International
  - Design of Reliable Communication Networks (DRCN 2016)
  - International Conference on Green IT Solutions (ICGREEN 2015)
- National
  - Algotel 2018 - 20eme Rencontres Francophones sur les Aspects Algorithmiques de Télécommunications
  - SDN DAY 2016 - Optimisation et algorithmes pour les réseaux SDN.
  - MOSIM 2014 - 10ème Conférence Francophone de Modéélisation, Optimisation et Simulation de l'économie linéaire à l'économie circulaire (session chair).

Workshop organization:

- SDN DAY 2016 - Optimisation et algorithmes pour les réseaux SDN.

Reviewer activity

Reviewer for International Journals:
Journal of Combinatorial Optimization, Mathematical Methods of Operations Research, Journal of Global Optimization, Computational Optimization and Applications, European Journal of Operations Research, Discrete and Applied Mathematics, Computers and Operations Research, IEEE Transaction on Networking.

7.7.2 Scientific and organizational responsabilities

- From 2016 (team constitution date). Deputy head (responsable adjoint) for OPTIMIST research group at Loria
- 2016. PI - PEPS Mirabelle MIND: Mixed Integer Nonlinear programming for membrane system Design. Project in collaboration with LRGP (UMR 7274).
- 2007-2011. Active partecipation to the management of a research project between Alcatel-Lucent and the OR group of Politecnico di Milano
- 2007. Organization committee for HYCON-EECI Graduate School on Control (Paris).
- 2006. Part of the organization of the HYCON network of excellence (EU, CNRS,IdF)

7.7.3 PhD Jury Committees and revision

PhD Thesis Jury Member

- 2015: Dalal Belabed, in Informatics, Telecommunication and Electronics from Université Paris 6. Title: Optimization of Virtual Networks: design and assessment (examiner).

- 2014: Truong Khoa Phan, Ph.D. in Computer Science from Université de Nice - Sophia Antipolis. Title: Design and Management of Networks with Low Power Consumption (examiner).

PhD Thesis Reviewer
For Italian PhD schools' rules, a stabilized researcher with a good expertize in a scientific field can be reviewer for a Ph.D. Thesis, after approval of the PhD school commision (no HDR is necessary).

- 2014: Inad Nawajah, Ph.D. in Mathematical Models and Methods in Engineering from Politecnico di Milano. Title: Bayesian Analysis of home care longitudinal counts data.

7.7.4 Invited talks

Rome, February 2018, ALGORITMI A COLAZIONE – Minicourse *An introduction to stochastic global optimization - How to make simple algorithms smart!*, Universitá di Roma "Tor Vergata", Rome, (Italy). This seminar was articulated in two lessons of 3 hours and it was part of a PhD school coursus.
Nancy, May 2016, 1st Symposium Mathematics for Decision and Discovery (M4D2), *Introducing implicit information in optimization methods: how to make smart a simple algorithm*, LORIA, (France).
Montréal, Juillet 2014 *Handling shared protection in telecommunication network optimisation* (with Giuliana Carello, Politecnico di Milano), GERAD (Canada).
Nancy, March 2013: *Optimization problems in ICT: energy-awareness and quality of service issues*, at LORIA, Ecole de Mines de Nancy (France).
Tours, February 2013: *Optimization problems in ICT: energy-awareness and quality of service issues* at Laboratoire d'Informatique, Ecole d'Ingénieurs Polytechnique de l'Université de Tours (France).
Milano, December 2006: *A global optimization approach to circle packing* at Dipartimento di Elettronica e Informazione, Politecnico di Milano (Italy).
Glasgow, November 2006: *Stochastic global optimization methods and an application to low-trust trajectory design* at Department of Aerospace Engineering, University of Glasgow (UK).
Evanston/Chicago, August 2004: *Trust region framework for global optimization* at Northwestern University, IL (USA).
Vienna, November 2004: *Trust region framework for global optimization* at Institut fur Statistik und Decision Support Systems, University of Vienn (Austria).

7.8 Additional professional activities

- June 2006-June 2008. External consultant for Giunti Multimedia s.r.l. (important italian publishing house). Design and development of optimization algorithms for book packing. The underlying optimization problem is a 3-dimensional bin-packing with different size boxes and side-constraints.

7.9 Teaching

In what follows OR stands for Operations Research.

From the arrival at UL-ENSMN (September 2013)

- Courses in English:
  - 2015/16-2016/17 OR (74 h/year)
  - 2013/2014 Exercise session for OR (2A, 34 h/year)
  - 2013/2014-2015/16 Discrete and deterministic optimization (2A, 31.5 h/year)
  - 2013/2014-2016/17 Advanced Discrete Optimization (2A, 31.5 h/year)
- Courses in French:
  - 2013/14-2014/15 Optimisation et Gestion de Production/Optimization and Production Managing (31.5 h/year)
  - 2014/15-2016/17 Cours d'introduction au Département GIMA/Introduction course to the GIMA Department (31.5 h/year)
- Tutoring for students projects 2A, 3A. Industrial and scientific projects.

Previous teaching activities

- 2008/09 OR Workshop course (master students) Politecnico di Milano
- 2005/06-2006/07 Combinatorial Optimization course (master students) at Univ. di Firenze
- Exercise session for different OR courses (master and bachelor):
  - 2011/12-2012/13 Universitá degli Studi di Torino
  - 2007/08, 2009/10 Politecnico di Milano
  - 2006/07 Universitá degli Studi di Siena
- Lab session for OR courses (master students):
  - 2008/2009-2010/11 Politecnico di Milano
  - 2001/2002, 2005/06-2006/07 Università degli Studi di Firenze
- Supervisor for bachelor and master thesis projects

8 Short summary of the Research Activity

My research mainly focused on:

- Combinatorial and MILP (Mixed Integer Linear Programming) applied to problems arising from ICT (Information and Communication Technology):
  - Network design and facility location problems:
    - heuristic methods for designing and routing of real-size network problems;
    - development of branch and price methods and heuristics for two-level facility locations;
    - study of formulations, bound and heuristics for network protection mechanisms;
    - exact and heuristic methods for energy-aware networking;
    - network virtual function placement and routing formulations and complexity results
  - Cloud Computing: heuristic methods for online resource management of cloud systems.
- Global optimization:
  - Stochastic methods based on random sampling and exploiting standard local search; construction of test functions for global optimization algorithms
  - Global Optimization methods applied to:
    - space mission analysis (development of global optimization algorithms for NonLinear and Mixed Integer problems in trajectory optimization)
    - molecular problems (molecular cluster optimization: theoretical and numerical optimization)
    - circle packing problems
    - critical node problem (constructive heuristics and theoretical results)
- MINLP (Mixed Integer Nonlinear programming) with application to membrane system design

Global Optimization

I focused my Ph.D. thesis on global optimization methods for unconstrained or

box constrained non linear optimization problems, that is:

$$\min_{x \in X} f(x)$$

with $f(\cdot)$ smooth, and $X = R^n$ or X equal to a box. The methods studied were based on random sampling and exploitation of standard local searches. The work of the thesis started from a biological problem: protein docking. The process of docking can be described as the geometrical configuration that two proteins attain when they are in close interaction. In particular given the energy interaction between the two proteins, the minimal configuration energy represents the docking configuration. Due to the complexity of the energetic landscape resulting, the optimization problem is very difficult, in fact the number of local but not global optima is quite huge, even with the simplified assumption that the proteins are rigid bodies. Even if the number of local optima is very high, the position of the local and global optima is believed to follow a particular pattern, that is called "funnel" structure. The general idea is that the landscape can be considered as the combination of a "simple" function with one or a few local minima and a "noise" that produce the large number of local minima. Therefore techniques to exploit these characteristics were developed and used ([O9],[J27],[J26, J25]).

My research on global optimization methods continued on some specific topics, such as circle packing ([J21, J20]), space trajectory design ([J19, J10]) and test functions for general non convex smooth problems ([J24]). The central elements of all these works is to embed in the solution algorithm some "a priori" knoledge about the structure of the problem and/or the global optima, even if this knowledge cannot always be expressed as an explicit mathematical property. For example, in protein docking the funnel structure of the objective function landscape is exploited using Monotonic Basin Hopping algorithm (that allow to move from one local optima to an lower adjacent one, creating a descending path along the funnel structure). I participated with the Global Optimization Laboratory of the University of Florence, in several international competitions for space trajectory design (GTOC competition) and in 2010, together with the ACT-ESA team (Advanced concept team, European Space Agency), we got the fourth place ([J10]). We participated and won an international optimization competition on circle packing problem (see [J20]). The competition problem was to find the smallest circle that can contain circles of increasing radius from 1 to n. This can be considered as a global continuous optimization problem with an underline combinatorial structure (due to the discrete values of the radii of the circles). We proposed a solution technique integrating the use of local searches and a simple combinatorial heuristic based on switching the position of similar size circles. To increase the speed of the optimization, the smallest circles are removed and re-inserted after the end of the global procedure.

I worked also on some theoretical issues in global optimization. One problem is the determination of minimal distances between particles in atomic clusters ([J23,

J18]). These bounds give an insight of the general structure of the global optima of these problems, and also can be considered an "a priori" information useful to be integrated in solution methods (such as continuous branch and bound techniques). Another topic, lying between global optimization and combinatorial optimization, is the critical node problem: given a graph, remove a subset of nodes of cardinality K to minimize the measure of the network connectivity. We proved complexity and inapproximability results for the problem where the number of residual connected component is minimized and we proposed a dynamic programming method to solve it ([J14]). To solve large size instances, we proposed a constructive heuristic that is able to improve all the state of the art putative optima, and give good results on very large instances ([J8]).

Combinatorial and MILP

Since 2007, I broaden my research interests to some topics of discrete optimization. In particular, for four years, I participated to an industrial research project in collaboration with Alcatel-Lucent. The aim of the project was to study and implement models and solution methods to estimate costs for network design. We studied routing and design problems in two layer networks ([J22]). Solution approaches have to take into account several technological features in order to provide solutions that can be useful to estimate costs for real networks. The development heuristic approaches have been implemented in order to be embedded within the network management framework developed by Alcatel-Lucent. The studied heuristics are based on the solution of routing sub-problems, with incremental costs, through ILP solvers.

Particular attention was given to study ILP models for routing problems with point-to-point dedicated and shared protection, taking into account the Shared Risk Group (SRG) concept. Both dedicated and shared protection ask for two paths to route each demand, a primary path and a backup one, to be used in case of failure. In dedicated protection the capacity of the backup path is fully dimensioned as for a primary path, on the contrary on the shared case, the backup paths of two different demands can share capacity if their primary paths are not affected by the same failures. The shared protection introduce non-linear constraints in the model, that can be linearized throw bottleneck constraints, leading to challenging problems even considering single demand routing with simple modular link dimensioning. SRGs generalize the concept of single link or node failure. A SRG is a set of links (and/or nodes) that fails for the same catastrophic event. Two paths that contain links (or nodes) that belong to the same SRG, are considered to fail simultaneously. The introduction of SRGs even with simple protection mechanism, like point-to-point dedicated protection, leads to problem that are NP-hard, if no particular structure is present in the topology of the network and/or SRGs. Routing problems with shared protection and SRG turn out to be very challenging. Thus we developed several alternative formulations, constraint elimination relaxations and MILP based heuristics in order to provide good lower and upper bounds for the problems ([C8],[O6, O6]).

In the contest of network design, I also worked on methods for solving two-level facility location problems, that are used to suitable model Fiber to the Home network problems. We proposed a an exact method based on Dantzing-Wolfe reformulation and discretization([J17]). We also developed heuristic strategies to solve larger dimension problems that combines local search, variable scale neighborhood search, very large scale neighborhood search and ILP based neighborhoods ([J15]).

Since 2010, I focused my research on energy-aware networking and cloud computing problems. In this context the research is two-fold: determine and validate suitable models for energy management and develop algorithms to efficiently solve these models.

We studied models for routing a set of demands with variable capacity requests during the day. The aim of optimization is to reduce energy consumption adapting the routing to allow the switch-off of node and arc devices, taking into account side constraints for device reliability. For this problem, we proposed energy models, heuristic approaches to deal with large size networks and an on-line procedure to determine the routing with updated predictions of traffic demand at a small time scale([C4],[C6]). The problem of dealing also with protection mechanisms was further investigated ([C5],[C7]). Furthermore, a robust approach based on dualization (Bertsimas and Sim, 2004) was proposed ([C4], [J12]). Due to the computational challenge introduced by protection mechanisms, reformulation and a branch and cut method based on such reformulation was proposed to solve larger size instances with protection ([J5],[J4]).

We studied and reorganized all the works on green networking for IP network, proposing a cataloging based on optimization modelling features ([J7]).

In the context of cloud computing, we studied the problem of online assigning a set of application requests to servers to obtain a trade off between energy consumption and performance. We propose a self-managing framework [C9] using hourly basis work load predictions. The framework can decide to shut-down or power up a server, or to move virtual machines (VMs) from one server to another by exploiting the VM live-migration mechanism, providing also availability guarantees to end-user applications. The resulting problem is a MINLP. We proposed a heuristic method to tackle the problem based on local search and diversification [C9], with a specific procedure to recover availability constraints. With the aim of dealing with large size clouds and to optimize the problem at a finer time scale, we propose a distributed hierarchical framework based on a mixed-integer non-linear optimization for online resource management across multiple time-scales [J16]. This framework allows to manage the cloud with good performances, allowing to follow demand variation through a process of re-optimization at the scale of few (5-15) minutes.

Due to the pervasive concept of virtualization, the new generation of Cloud systems can be based on a strict integration of service centers and networking infrastructures. This phenomenon gives rise to problems taking into account demand routing

and network dimensioning with assignment/scheduling problems (typical of cloud computing).

The geographical distribution of the computing facilities offers many opportunities for optimizing energy consumption and costs by means of a clever distribution of the computational workload exploiting different availability of renewable energy sources, but also different time zones and hourly energy pricing. Energy and cost savings can be pursued by dynamically allocating computing resources to applications at a global level, while communication networks allow to assign flexibly load requests and to move data. We propose an optimization framework able to jointly manage the use of brown and green energy in an integrated system and to guarantee quality requirements. We propose an efficient and accurate problem formulation that can be solved for real-size instances in few minutes to optimality ([J13]).

Another application arising from the integration of cloud and network is the Network Functions Virtualization, that allows to provide network services by chaining multiple Virtual Network Functions (VNFs) that are executed on commodity servers located in nodes distributed across the network. Demands must be routed in the network, and pass by the nodes where the requested VNFs are installed. The resulting optimization problem can be view as the integration of a facility location problem (to decide where to install VNFs) and a routing problem (to route all the demands in such a way that they pass by all the requested VNFs). We defined a generic VNF Placement and Routing (VNF-PR) optimization problem, formulate it as a MILP problem, including additional constraints important from the application point of view: compression/decompression of demand flows, maximal demand latency (introduced by links and VNFs), etc. In order to reduce the computational time needed for solving the problem, we designed a math-heuristic, integrating a dicothomic search and a sequential resolution of different models, starting from a simplified version of the model to the complete one ([C2], [J2]).

We studied a simplified version of the problem, namely one single VNF to locate. For this version we determined computational complexity on different capacity scenarios and network topologies ([J3]). We presented a revised version of the two mathematical programming formulations proposed in the literature to address the problem, compare them theoretically and numerically. Furthermore, we propose additional cuts and evaluate their impact on the linear relaxation and on the full model ([W1]).

MINLP for Membrane System Design

From 2015, I started a research collaboration with the LRGP (UMR7274) for the development of an optimization tool dedicated to membrane system design for gas separation. The collaboration was extended to the international with the integration of Veronica Piccialli (University of Rome "Tor Vergata"). The aim of the collaboration is to put together two complementary expertise: the deep knowledge about membrane systems (LRGP) and mixed integer non linear optimization

(LORIA/Rome). From an optimization point of view, the design of membranes can be modeled as a Mixed Integer Nonlinear Programming (MINLP) problem. Continuous variables and constraints allow to model a single membrane behavior, and discrete variables are necessary to parameterize the number and connection of different membranes (and eventually the type of membrane involved). The collaboration started with a stage at LRGP/LORIA and a master thesis titled "An optimization framework for membrane system automatic parameter tuning" at University of Rome, under the supervision of Veronica Piccialli (Rome) and myself. In the master thesis a first optimization model able to solve a fixed design membrane system in a simplified form has been proposed and validated and was presented in a conference [O1]. The collaboration is continuing. In 2016 we received the support of a PEPS Mirabelle (MIND) and obtained a grant for Ph.D. thesis (Marjan Bozorg) started in October 2016. The PhD thesis is part of an international agreement for a double title between the University of Lorraine and the University of Rome "Tor Vergata". M. Bozorg obtained a Vinci grant for mobility between France and Italy. The proposed method was validate using a case study from the literature, and used to make a through study for the CO_2 membrane capture for steel production industries ([J1]).

Other research activity

Part of my research was devoted to other topics:

- For two years (2006-2008), I was external consultant for Giunti (one of the most important Italian publishing houses). The central part of the work was to study and implement algorithms for automatically determine the packing of book orders for delivery. The problem is a three-dimensional packing with containers of different sizes and with several side constraints: the books in each container have to respect a given orders, the possibility of choosing a given container is dependent on the presence of particular items. An algorithm based on the generalization of the bottom-left strategy for 3-dimensional packing, integrated in a local search approach based on subproblems re-optimization was implemented and tested on real instances.
- From 2010 to 2013, I collaborated in a national funded project on health care management, focused on personal and resource management from a patient centered perspective. Part of the work was devoted to personal scheduling problems. Among several problems, we considered the case study of the emergency call center of the city of Milan (118 service). We developed different models with the aim of optimally compose teams to guarantee minimal response time for the users of 118 ([B2],[O8, O7]). A part of the work was devoted to operating room planning problems and continued after the conclusion of the project. In a recent study, we started to investigate the introduction of a robust approach to the assignment of patients to operating room blocks to take into account stochastic surgical durations ([O4],[C3]). Then, we ex-

tended the robust model to a rolling horizon approach taking into account cancellations and new arrivals ([J6]).

- From 2013 to 2016, I participate to the Regional project Satelor and co-direct a PhD thesis on wireless sensor networks with the aim of developing models and algorithms to manage energy consumption and guarantee adequate levels of Quality of Service. We proposed a new probabilistic protocol to guarantee a balanced usage of the sensors, allowing to increase the expected lifetime of the overall network ([C1]).

9 List of publications

In the operations research community is common practice to list the authors in alphabetical order, therefore almost all my contributions follow this rule. Exceptions to this rule are explained by different policies of different communities (where, for example, the first author must be the PhD student working on the project to highlight her/his original contribution), or specific circumstances:

- in [J2],[J1] and [C1], a PhD student and/or a young Post-doc are the first authors;
- in [J10], as the paper presents the results of a large team work on an optimization competition, we decided for a first arrival first served policy;
- in [J23], W. Schachinger gave a major contribution to the work (the other authors follow again the alphabetical order rule)

Summary
- International Journals: 27
- Book chapters: 2
- International Conferences with review: 9

Google scholar: `https://scholar.google.com/citations?user=viE0O9MAAAAJ&hl=en`
DBLP: `https://dblp.uni-trier.de/pers/hd/a/Addis:Bernardetta`

Working papers

[W1] B. Addis, G. Carello, F. De Bettin, and M. Gao. On a network virtual function placement and routing problem: properties and formulations. *Draft version available on HAL*, 2018.

International Journals

[J1] Á. A. Ramírez Santos, M. Bozorg, B. Addis, V. Piccialli, C. Castel, and E. Favre. Optimization of multistage membrane gas separation processes. Example of application to CO2 capture from blast furnace gas. *Journal of Membrane Science*, 566:346–366, 2018.

[J2] M. Gao, B. Addis, M. Bouet, and S. Secci. Optimal orchestration of virtual network functions. *Computer Networks*, 142:108–127, 2018.

[J3] B. Addis, M. Gao, and G. Carello. On the complexity of a virtual network function placement and routing problem. *Electronic Notes in Discrete Mathematics*, 69:197–204, 2018.

[J4] B. Addis, G. Carello, and S. Mattia. Survivable green traffic engineering with shared protection. *Networks*, 69:6–22, 2017.

[J5] B. Addis, G. Carello, and S. Mattia. Energy-aware survivable networks. *Electronic Notes in Discrete Mathematics*, 52:133–140, 2016.

[J6] B. Addis, G. Carello, A. Grosso, and E. Tànfani. Operating room scheduling and rescheduling: a rolling horizon approach. *Flexible Services and Manufacturing Journal*, 28(1):206–232, 2016.

[J7] B. Addis, A. Capone, G. Carello, L.G. Gianoli, and B. Sansò. Energy management in communication networks: a journey through modeling and optimization glasses. *Computer Communications*, 91-92:76–94, 2016.

[J8] B. Addis, R. Aringhieri, A. Grosso, and P. Hosteins. Hybrid constructive heuristics for the critical node problem. *Annals of Operations Research*, 238(1):637–649, 2016.

[J9] B. Addis, G. Carello, A. Grosso, E. Lanzarone, S. Mattia, and E. Tànfani. Handling uncertainty in health care management using the cardinality-constrained approach: Advantages and remarks. *Operations Research for Health Care*, 4:1–4, 2015.

[J10] D. Izzo, L. F. Simões, C. H. Yama, F. Biscani, D. Di Lorenzo, B. Addis, and A. Cassioli. GTOC5: Results from the european space agency's advanced concepts team and university of florence global optimization laboratory. *Acta Futura*, 8:45–55, 2014.

[J11] B. Addis, A. Capone, G. Carello, L.G. Gianoli, and B. Sansó. Energy management through optimized routing and device powering for greener communication networks. *IEEE/ACM Transactions on Networking*, 22(1):313–325, February 2014.

[J12] B. Addis, A. Capone, G. Carello, L. Gianoli, and B. Sansò. On the energy cost of network protection techniques. *Computer Networks*, 75:239–259, 2014.

[J13] B. Addis, D. Ardagna, A. Capone, and G. Carello. Energy-aware joint management of network and cloud infrastructures. *Computer Networks*, 70:75–95, 2014.

[J14] B. Addis, M. Di Summa, and A. Grosso. Removing critical nodes from a graph: complexity results and polynomial algorithms for the case of bounded treewidth. *Discrete and Applied Mathematics*, 161:2349–2360, 2013.

[J15] B. Addis, G. Carello, and A. Ceselli. Combining very large scale and ILP based neighborhoods for a two-level location problem. *European Journal of Operational Research*, 231:535–546, 2013.

[J16] B. Addis, D. Ardagna, B. Panicucci, M. Squillante, and Zhang L. A hierarchical approach for the resource management of very large cloud platforms. *IEEE Transactions on Dependable and Secure Computing*, 10:253–272, 2013.

[J17] B. Addis, G. Carello, and A. Ceselli. Exactly solving a two-level hierarchical location problem with modular node capacities. *Networks*, 1:161–180, 2012.

[J18] B. Addis and W. Schachinger. Improved bounds for interatomic distance in morse clusters. *Operations Research Letters*, 37:290–294, 2009.

[J19] B. Addis, A. Cassioli, M. Locatelli, and F. Schoen. A global optimization method for design of space trajectories. *Computational Optimization and Applications*, 2009.

[J20] B. Addis, M. Locatelli, and F. Schoen. Efficiently packing unequal disks in a circle. *Operations Research Letters*, 36:37–42, 2008.

[J21] B. Addis, M. Locatelli, and F. Schoen. Disk packing in a square: a new global optimization approach. *INFORMS Journal on Computing*, 20:516–524, 2008.

[J22] B. Addis, A. Capone, G. Carello, F. Malucelli, M. Fumagalli, and E. Pedrinelli. Designing two-layer optical networks with statistical multiplexing. *Fiber and Integrated Optics*, 27:249–255, 2008.

[J23] W. Schachinger, B. Addis, I. M. Bomze, and F. Schoen. New results for molecular formation under pairwise potential minimization. *Computational Optimization and Applications*, 38:329–349, 2007.

[J24] B. Addis and M. Locatelli. A new class of test functions for global optimization. *Journal of Global Optimization*, 38:479–501, 2007.

[J25] B. Addis and S. Leyffer. A trust-region algorithm for global optimization. *Computational Optimization and Applications*, 35:287–304, 2006.

[J26] B. Addis, M. Locatelli, and F. Schoen. Local optima smoothing for global optimization. *Optimization Methods and Software*, 20:417–437, 2005.

[J27] B. Addis and F. Schoen. Docking of atomic clusters through nonlinear optimization. *J. of Global Optimization*, 30:1–21, 2004.

International Conferences with revision

[C1] E. Tsiontsiou, B. Addis, A. Ceselli, and Y. Song. Optimal probabilistic energy-aware routing for duty-cycled wireless sensor networks. In *Proc. of NTMS 2016, Mobility and Wireless Track*, 2016.

[C2] B. Addis, D. Belabed, M Bouet, and S. Secci. Virtual network functions placement and routing optimization. In *Proc. of IEEE CLOUDNET 2015*, 2015.

[C3] B. Addis, G. Carello, and E. Tànfani. A robust optimization approach for the operating room planning problem with uncertain surgery durations. In *Proceedings of the International Conference on Health Care Systems Engineering*, volume 61. Series: Springer Proceedings in Mathematics & Statistics, 2014.

[C4] B. Addis, A. Capone, G. Carello, L. Gianoli, and B. Sansò. A robust optimization approach for energy-aware routing in mpls networks. In *Proceedings of IEEE ICNC 2013*, January 2013.

[C5] B. Addis, A. Capone, G. Carello, L. Gianoli, and B. Sansò. Multi-period traffic engineering of resilient networks for energy efficiency. In *Proceedings of IEEE GreenCom 2012*, September 2012.

[C6] B. Addis, A. Capone, G. Carello, L. Gianoli, and B. Sansò. Energy-aware multiperiod traffic engineering with flow-based routing. In *Proceedings of IEEE ICC 2012 (workshop on Green Communications and Networking - GCN)*, pages 5957–5961, 2012.

[C7] B. Addis, A. Capone, G. Carello, L. Gianoli, and B. Sansò. Energy aware management of resilient networks with shared protection. In *Sustainable Internet and ICT for Sustainability (SustainIT), 2012*, pages 1–9, 2012.

[C8] B. Addis, G. Carello, and F. Malucelli. SRG-disjoint design with dedicated and shared protection. In *Lecture Notes in Computer Science*, pages 18–23, 2011.

[C9] B. Addis, D. Ardagna, B. Panicucci, and L. Zhang. Autonomic management of cloud service centers with availability guarantees. In *Proc. of the 4th IEEE International Conference on Cloud Computing*, pages 220–227, 2010.

Book chapters

[B1] B. Addis, R. Aringhieri, E. Tànfani, and A. Testi. Clinical pathways: Insights from a multidisciplinary literature survey. In *Proceedings of the 38th conference of the EURO working group Operational Research Applied to Health Services (ORAHS).* CHOIR: Center for Healthcare Operations Improvement & Research, of the University of Twente., 2012.

[B2] B. Addis, R. Aringhieri, G. Carello, A. Grosso, and F. Maffioli. Workforce management based on forecast demand. In *Advanced Decision Making Methods Applied to Health Care, 2012. International Series in Operations Research & Management Science*, volume 73, pages 1–11. Springer, 2012.

Other publications

[O1] M. Krzaczkowska, B. Addis, R. Bounaceur, E. Favre, A. Oulamara, and V. Piccialli. An optimization framework for membrane system automatic parameter tuning. In *17ème conférence de la Société Franccaise de Recherche Opérationelle et Aide à la Décision*, 2016.

[O2] B. Addis, C. Carello, and M. Raco. Heuristics for an energy-aware management problem in cellular networks. In *16ème conférence de la Société Franccaise de Recherche Opérationelle et Aide à la Décision*, 2015.

[O3] B. Addis, C. Carello, and G. La Rosa. An energy-aware management and design problem in wireless cellular networks. In *International Network Optimization Conference 2015*, 2015.

[O4] B. Addis, G. Carello, and E. Tànfani. A robust optimization approach for the operating room planning problem with uncertain surgery durations. In *14ème conférence de la Société Franccaise de Recherche Opérationelle et Aide à la Décision*, 2013.

[O5] B. Addis, G. Carello, and F. Malucelli. Srg-disjoint design with dedicated and shared protection. In *International Network Optimization Conference 2011*, 2011.

[O6] B. Addis, G. Carello, and F. Malucelli. Network design with SRG based protection. In *Proc. of the 10th Cologne-Twente Workshop on graphs and combinatorial optimization. Extended Abstracts*, 2011.

[O7] B. Addis, R. Aringhieri, M. Gribaudo, and A. Grosso. Combining petri nets and metaheuristics for the optimal composition of medical teams. In *Proceeding 9th Metaheuristics International Conference*, 2011.

[O8] B. Addis, R. Aringhieri, G. Carello, M. Gribaudo, and A. Grosso. Advanced workforce management in healthcare. In *Proceedings ORAHS 2011*, pages 308–319, 2011.

[O9] B. Addis. *Global optimization using local searches*. PhD thesis, Computer Science and Automation Engineering, 2005.